MASTER IN COMPUTER SCIENCE

Storage Management Subsystem Environment in Front of Device Performance

Parmentier, Marc

Award date:
1991

Awarding institution:
University of Namur

FACULTES
UNIVERSITAIRES
N.D. DE LA PAIX

NAMUR

INSTITUT D'INFORMATIQUE

STORAGE MANAGEMENT SUBSYSTEM ENVIRONMENT IN FRONT OF DEVICE PERFORMANCE

par Marc Parmentier

Promoteur :
Professeur J.Ramaekers

Mémoire présenté en vue
de l'obtention du titre
de Licencié et Maître
en Informatique

Année académique 1990-1991

rue Grandgagnage 21B 5000 Namur (Belgium)

First of all, I would like to express my gratefulness to Mr Cardinael and Mr Moerman from the 'ASLK-CGER Bank and Insurance' who has offered me the opportunity to accomplish such an interesting training.

I would like to express my thankfulness to Mr Maes who has devoted a lot of his precious time to the follow up of this work, his practical experience as well as his pedagogical skills were extremely helpful.

Secondly, I wish to thank all the people of the CTI from the 'CGER' for the support they have brought during the training period in Brussels.

Next, I would like to express my thankfulness to Mr. Ramaekers, professor of the 'Institut d'informatique' of the University of Namur who has consented to guide this work.

Finally, I would like to thank Mr. H. Pat Artis from 'Performance Associates Inc', for the precious discussion and the support he has provided through the completion of this work.

Abstract

The minimisation of the Direct Access Storage Device response time, which has an effect on both system throughput and end-users, is the primary objective of the storage administration. Continuing growth in the size and complexity of large systems has created a need to simplify and automate storage management functions, and to ensure the performance requirements. Under MVS, IBM suggested a solution called 'Storage Management Subsystem'. The goal of this end-study-paper is to introduce and present the heaviness of performance detection problems, to depict and critic this subsystem solution, and to suggest major point to highlight before and during this subsystem implementation in order to improve its good working.

La minimisation du temps de réponse des disques, influence le fonctionnement du système et, c'est l'objectif premier de la gestion des mémoires. La croissance continue en taille et en complexité des grands systèmes a créé un besoin de simplification et d'automatisation des fonctions de gestion des supports mémoires. Sous MVS, IBM a suggéré une solution appelée 'Storage Management Subsystem (SMS)'. L'objectif de ce travail de fin d'études est d'exposer et de mettre en évidence la lourdeur des problèmes d'analyse des performances, de décomposer et de critiquer la solution SMS, et, à partir de là suggérer les points importants à traiter avant et pendant l'implémentation de ce sous-système pour en améliorer le fonctionnement.

Preface

This paper could be cluttered with footnotes at the mention of each term that is a trademark if each citation has to be given in full.
To avoid this kind of footnotes, the following section suggest the noticed trademarks used within this paper.

MVS/ESA™, MVS/DFP™, DFSMS™, DFHSM™, DFSORT™, DFDSS™, RACF™, RMF™, SMF™, REXX™, ISPF™, ISMF™ are trademarks of the International Business Machines Corporation.

ASTEX™ and MICS™ are trademarks of LEGENT corporation.

SAS™ is a trademark of the SAS Institute.

Contents

Introduction

While we have to ensure a certain service level requirements within a huge system, the Disk Access Storage Device (DASD) is a major resource.

The reader has to be warned that this all paper is based on a IBM 3090 under MVS/ESA environment.

In the first chapters we will depict the major components of a 'storage hierarchy', finally showing the impact of a good DASD subsystem working properly and the importance of the I/O boundary. Generalities concerning some tuning methods will be explained. A personal proposal, made during a training period at the "ASLK-CGER Bank and Insurance", over a performance problems detection tool based on thresholds bypassing will be depicted. Practical studies will in a last step be suggested.

As the disk storage resource size is unceasing growing, and the disk storage resource management requirements are also increasing, it is not possible any more to manage performance problems and others manually.

In chapter 4, we will depict the Storage Management Subsystem (SMS) concepts and constructs. These concepts consist to suggest a dataset oriented management, and to separate the storage management into a logical and a physical domain.

Due to these key concepts, in the next chapters we will depict a new storage hierarchy understanding, we will also explain some SMS behaviours concerning performance and critic them.

As a performance focus, we will explain how SMS takes the user service level requirements into account by the allocation and the open process. We will explain the limitation of the available performance tools, and suggest a new concept of data collector. Data collector enables to collect interpretable measures at a logical and physical level.

As the automatic allocation process is among others based on theorical behaviours. As no trustworthy control tools are not available to verify the 'datases' performance in all cases. We will suggest a pooling concept inforcing SMS to respect the pre-defined performance requirements, and explain the major pre-condition for a good SMS migration.

1. Elements of Storage Hierarchy

"An information-processing system is a set of hardware and software components capable of processing data according to user-written programs." [COH89]

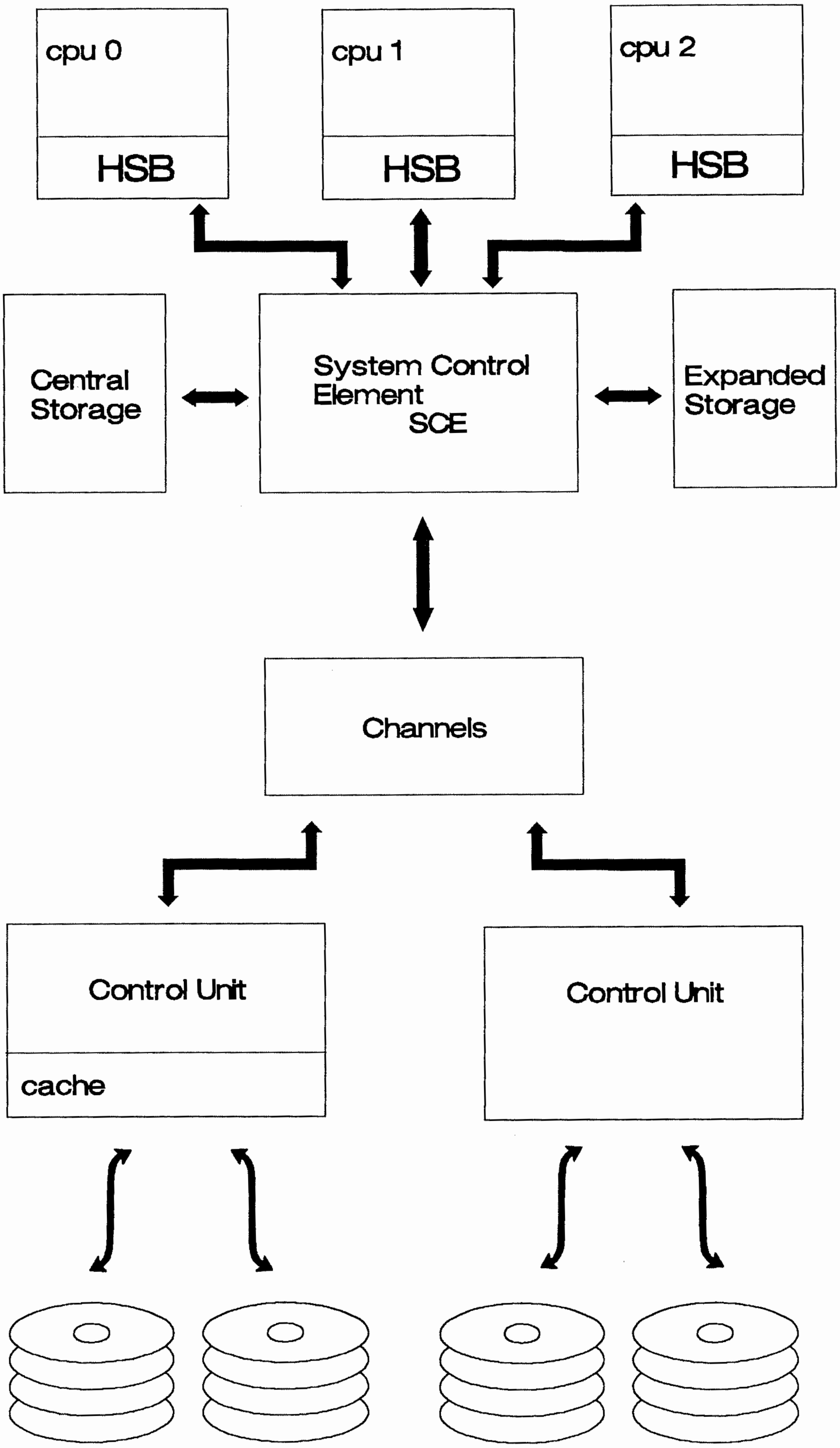
With the continued introduction of more powerful processing systems based on faster, smaller, and more reliable technology, the requirements on storage subsystems become more and more stringent. There is also a greater focus on storage cost and performance ratio, since storage is a determinant key of total system cost, throughput capacity, and responsiveness.

The types of available technology chosen which taken together meet the diverse requirements placed on the storage subsystem, consist of a hierarchy of storage levels. The significant new development in this hierarchy is the increasing importance of processor storage in diverting activity from the lower levels of the storage hierarchy, thus *reducing system I/O rates*. Increased usage of processor storage can be quick and orderly, by extending the existing concept of virtual storage and addressability.

The storage must be configurable into almost unlimited sizes to meet the data capacities required by ultra-large processing systems. Other factors such as cost, reliability, power, space, and cooling must also be reasonable. These items, however, are not discussed.

If we had to specify the ideal storage, we might imagine that the requirements would lead naturally to a single-level, monolithic store consisting of only one type of technology. The resulting store might be like a shared processor buffer of near-infinite size.

Figure 1.1



Direct Access Storage Device (DASD)

Such a storage structure would allow full realization of a processor's potential capacity, with no penalty for storage delays. In reality there is no single technology that can provide *speed*, *capacity* and *low cost*. There are a wide variety of storage technologies available which cover a wide range of performance. Faster technologies are usually more expensive and more difficult to configure in bulk.
Therefore, the best approach is to use the available options, taking advantage of the best features of each. This way, we can attempt to meet the full set of storage objectives as closely as possible.

The goal of this chapter is to explain generalities over the storage hierarchy components merely in regard of their performance and to show the importance of the Direct Access Storage Device.

1.1 Storage hierarchy data flow

The physical structure of the next described storage hierarchy can be viewed in the following figure (Figure 1.1), which shows a simplified diagram of a system. The diagram shows three CPUs, each with an integrated processor buffer which is also known as the High-Speed Buffer (HSB). The System Control Element (SCE) is the central and most complex component in this structure, because it routes data among all the CPUs, Central Storage, Expanded Storage, and all channels, while ensuring that elements are correct and up-to-date. Direct Access Storage Device (DASD) under cached or non-cached Control Units.

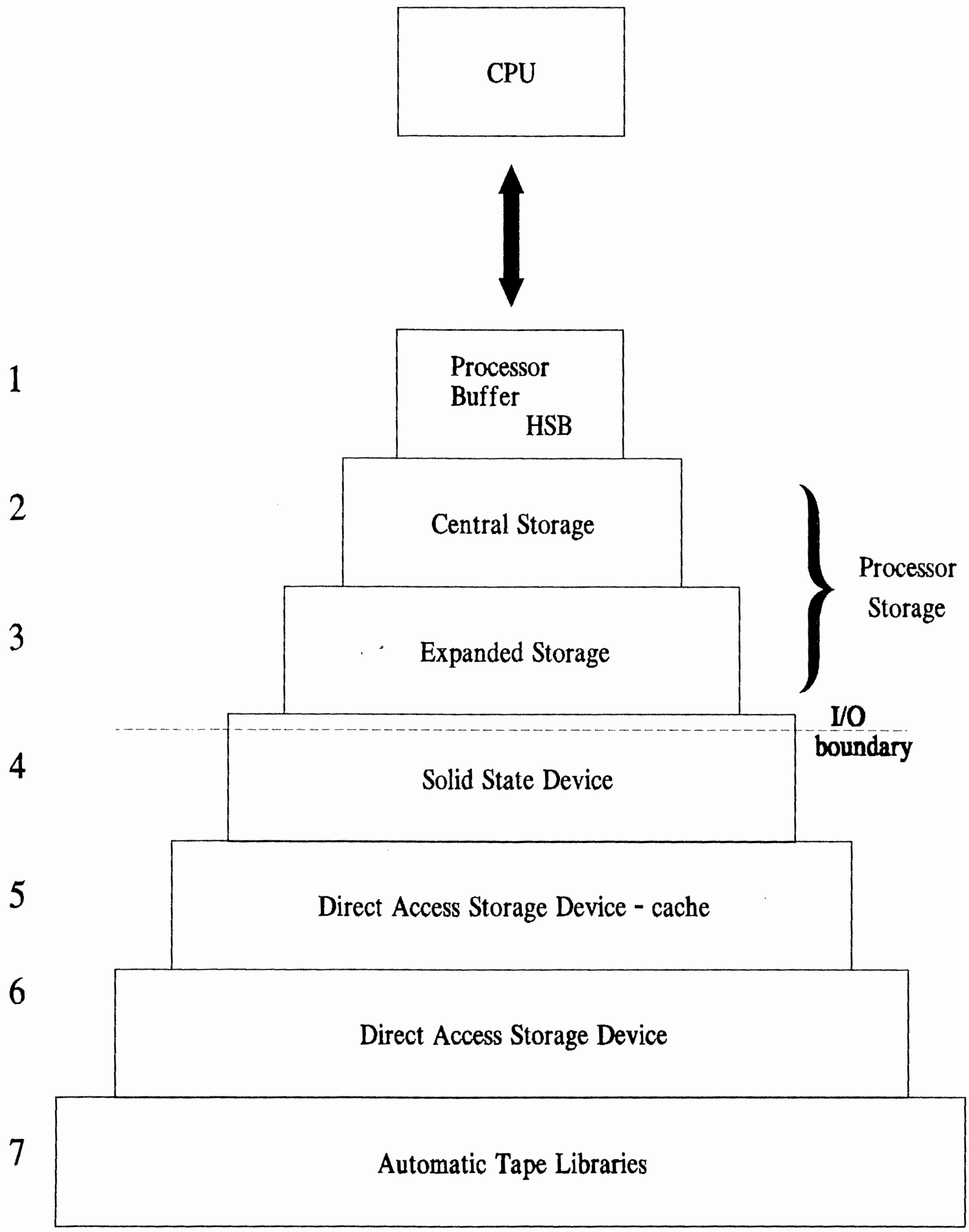
The flow of data through the physical components of the hierarchy can be best understood by the sequence of actions taken when a CPU accesses data elements resident in each storage type. The best performance case is a buffer hit, because the data element can be transferred directly to the appropriate CPU component in nanoseconds range.

If the data element is not in the buffer, it must be retrieved from the Central Storage through the SCE. These buffer misses typically take hundreds of nanoseconds to resolve, but the specific data element requested can be routed to the CPU component and in parallel, the entire set from which the data belongs to is transferred to the buffer.

A *page fault* occurs when the data element is not resident in Central Storage. With the advent of Expanded Storage technology, many page faults can now be satisfied in the tens of microseconds needed to move a 4K-byte block from the Expanded Storage to the Central Storage via the System Control Element (SCE). If the data element is not resident in either Central Storage or Expanded Storage, it must be retrieved from one of the system paging datasets via a READ I/O. In this case, the data element is transferred from the DASD through the control unit to the channel. The channel transfers the data to central storage via the SCE. At this point, the data are brought into the buffer of a CPU via a buffer miss. The next time the requesting program is scheduled to execute.

Data accessed from non-paging DASD on a typical I/O request follows the same path as that of the paging case previously described. In either case, the I/O can take tens of milliseconds if the data need to be retrieved from the DASD. In some cases, data can be retrieved much faster if present in a control unit cache. In this case, the typical DASD doesn't need to be accessed, and the requested data can be transferred directly from the electronic storage in the control unit in about 2-3 milliseconds. An other solution to retrieve data as much faster is putting the dataset in a Solid State Device. A Solid State Device is equivalent to a non volatile electronic storage. These both solution are logically equivalent on finding the data one level higher in the hierarchy on a faster but smaller capacity technology. This physical hierarchy can vary depending on machine type and system configuration, but when *viewed conceptually* we can study its general features.

Figure 1.2



Storage hierarchy based on :

. Access speed

. Capacity

. Cost

1.2 Element of storage hierarchy

In designing a storage system consisting of multiple heterogeneous technologies, each technology tries to take advantage of the key values of the other.

This storage system structure is conceptually viewed as a pyramid of levels as developed in the following figure (Figure 1.2). The levels are numbered sequentially starting from level 1 (L1), as we descend the hierarchy. The CPUs are positioned above the top level of the hierarchy because data elements are brought to the top level of the hierarchy for the CPU to access them.

The key characteristics in understanding the concept of a storage hierarchy are ACCESS SPEED, CAPACITY, and COST. As we move downward through the hierarchy, each subsequent level is slower and less expensive and usually configured in larger cost-effective capacities. The overall objective of the storage hierarchy is to provide average access speed almost as fast as the fastest level (HSB) with an average cost per bit of on-line data almost as low as the least expensive level. This can be achieved only if the vast majority of on-line data are resident on DASD while almost all of the CPUs storage accesses are satisfied from high-speed buffers.

Data elements are moved upward and downward through the storage hierarchy based on access and reused frequency. More often a data element has to be accessed and reused, more often its locality of reference allow a very high hit ratio in the HSB. Each level of the hierarchy is managed by a system component (hardware, software, or microcode). The management approach is generally based on a Least Recently Used (LRU) algorithm for replacing an inactive data element by a more active element. The element replaced is moved downward in the hierarchy, while the requested element is rising to the top of the hierarchy.

Given its primary position in the hierarchy, the performance of the HSB has a significant impact on the speed of the processor and the overall system performance. Levels 2 and 3 together constitute the processor storage (PS). Central Storage (CS) is the byte-addressable main storage of the processor system, and Expanded Storage (ES) is the 4K-byte block addressable extension to the central storage. The DASD subsystem is made up of DASD, cached DASD and Solid State Device. As the home location of the bulk of permanent on-line data, the DASD subsystem full fills the role of the cost effective bulk storage in the hierarchy. The DASD cache can be very effective in reducing the average response time and in reducing the access rate to the DASD. The Solid State Devices are also effective because they are always at least equivalent in average response time as read hit in DASD cache. Levels 7 constitute the tape storage. Tape storage are used for two specific purpose. The first one as a second hand support for less used data and the second one as archival storage.

Each of these elements are discussed in further detail in the following sections.

1.2.1 Processor high-speed buffer

The top level of the storage hierarchy is the processor buffer, which is also called the High-Speed Buffer (HSB). It is the most critical element in the hierarchy, because it satisfies processor storage references (both instructions and data) directly to the CPU logic elements and registers. The HSB performance, therefore, has a direct and powerful impact on processor performance and system capacity. The processor buffer needs to operate in the same range of speed as the processor to satisfy its storage references without causing any CPU delays. This speed requirement also means that the buffer must be physically close to the processor logic to minimize propagation delays. The L1 cache is typically tens to low hundreds of kilobytes in size (10-100_Kb), because of the high cost and spatial limitations. Processor buffers in this size range are extremely effective, because of the strong locality of reference exhibited by a program's instruction and data reference sequences. [COH89]

When a referenced data element is not found in the buffer, it must be retrieved from central storage. This is a much slower access, and the processor may become idle while waiting for a data element to be delivered. In this case, the requested data element is routed to both the processor logic and the buffer in parallel, to minimize this delay.

The effectiveness of a processor buffer is usually described by a buffer hit ratio, which is calculated as the percent of total storage references satisfied directly from the buffer with no access to central storage. Although the typical buffer hit range seems to provide high buffer performance, damages within the range are very important to CPU performance. Although there seems to be little difference between 98 and 96 percent buffer hit ratios. Another way to look over this difference is to say that the buffer miss ratio increased from 2 to 4 percent. This means that twice as much accesses to central storage are required.

Looking at it from the viewpoint of buffer misses gives a truer picture of the impact of buffer performance on total processor performance.

There are two distinct management philosophies for processor buffers based on when updates to data are pushed through in central storage. In a *store-through design*, updates are made to the buffer and to the corresponding home location of the data in central storage. In a *store-in design*, updates are made only when the data element is removed from the buffer by LRU replacement. The advantage of the store-through design is its simplicity. All updates are made to Central Storage immediately, which makes the newly changed data available to other processors from its Central Storage home location.
A store-in design has the advantage of reducing transfers between buffer and Central Storage, because changes are written back to Central Storage only when they are absolutely required. The controls required for store-in are more complex than store-through, particularly in tightly coupled CPUs. In this case, a storage reference on one processor may need to be checked against the contents of buffers on other CPUs to determine whether it contains an updated version of the data. Overall, one cannot say wether one approach is superior, because much depends on the objectives of the design. Either design may be the best solution, depending on the set of wished criteria. [COH89]

1.2.2 Processor Storage

Processor Storage includes the SYNCHRONOUS LEVELS of the storage hierarchy. That is, the processor waits while an item of data is retrieved from a level of this class of storage. Processor Storage is closely integrated with the processor, and the time to retrieve an item from it is measured in microseconds or less. Because this access time is so fast at this level of the storage hierarchy, it is more efficient for the processor to pause momentarily to wait for an item to be retrieved from Processor Storage than to have the processor searching for another task to execute.

The size of processor storage generally ranges from tens of megabytes to several gigabytes and consists of multiple 4096-byte blocks, called *frames* of processor storage. The contents of processor storage are managed by the operating system. Through the use of a global LRU scheme, the operating system tends to keep the most actively referenced programs and data areas of the system users in processor storage.

There are two types of processor storage in the storage hierarchy: *Central Storage* and *Expanded Storage*, the characteristics of which are discussed in the following sections.

CENTRAL STORAGE

Central storage (which is often called real storage) is *byte-addressable* processor storage. For each addressed frame of central storage, the hardware maintains reference, change, and access key indicators in special high-speed storage packaged close to the processors. These indicators are used by the operating system to preserve data integrity and to assist in its LRU management of central storage.

Although conceptually sitting between two levels of the storage hierarchy, central storage is physically accessible to three levels:

. the high-speed buffer (HSB) of each processor,

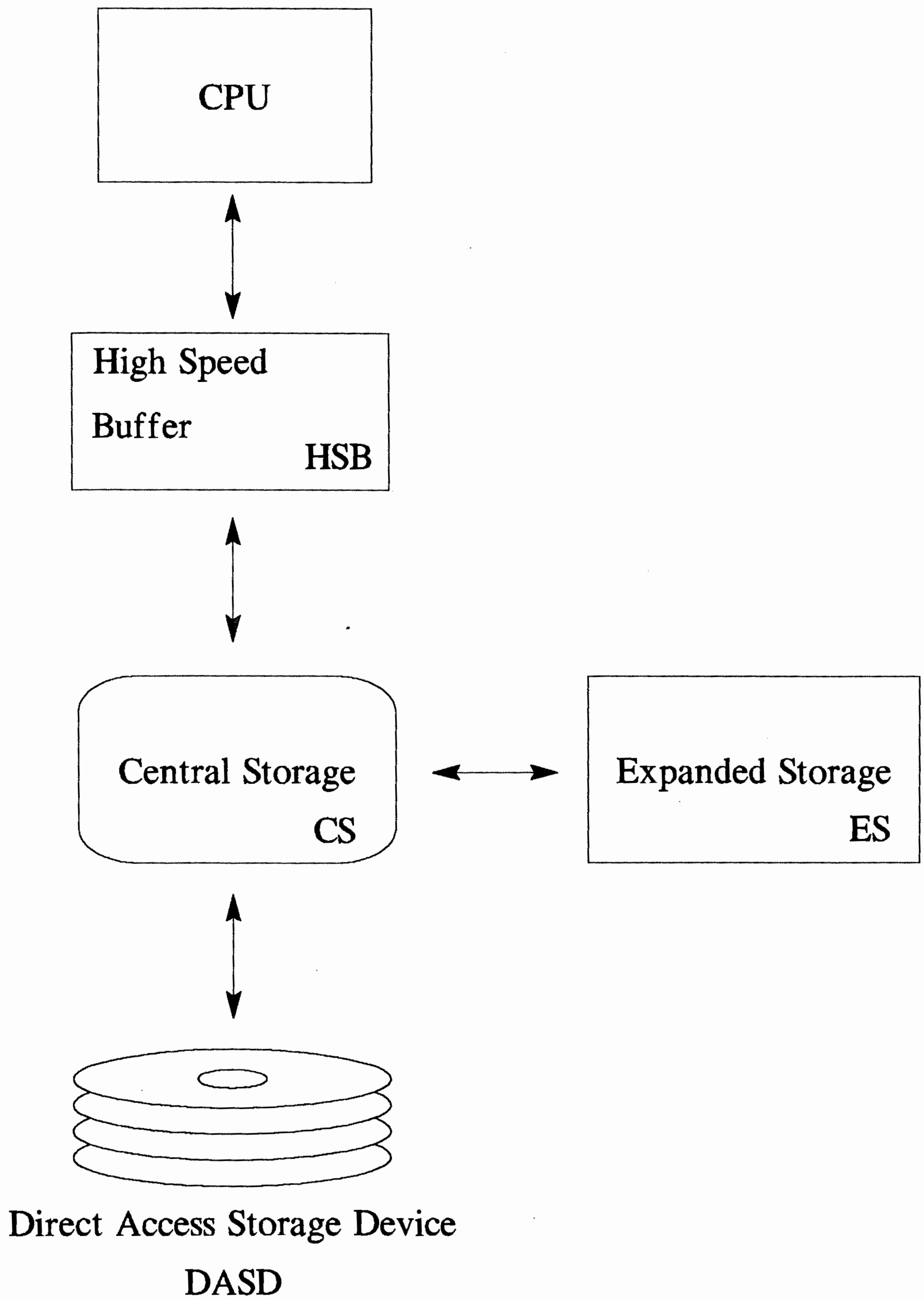
. the expanded storage,

. the I/O subsystem.

When a processor requires a data item that is located in central storage, the few bytes of central storage that contains the item is moved into its HSB. This operation is managed by the hardware and a fraction of microsecond to complete. Transfers between Central Storage and Expanded Storage are accomplished in frame-size pieces under the control of the operating system in times measured in tens of the microseconds. Applications usually originate requests for data movement between Central Storage and the DASD subsystem. These moves are made through a channel interface in times measured in milliseconds.

The operating system keeps the most active set of programs and data areas in central storage. Generally, items can remain unreferenced in central storage for tens of seconds before they are moved out to a lower level of the hierarchy.

Figure 1.3



Data flow within the
Storage hierarchy

EXPANDED STORAGE

Expanded Storage is *block-addressable* Processor Storage. An address generated for expanded storage points to a 4096-bytes frame of expanded storage. The operating system maintains information in a control block for each frame of expanded storage, and that information is used for preserving data integrity and for LRU management of expanded storage.

Conceptually, expanded storage sits between the central storage and the DASD subsystem levels of the storage hierarchy. However, expanded storage is physically accessible only through central storage. When a processor requires a data item that is located in expanded storage, the contents of the expanded storage frame containing the item are moved into a central storage frame. This operation is managed by the operating system and takes tens of microseconds to complete. Once the data resides in central storage, it is then moved into the processor's HSB, as already described. This data flow within the Storage Hierarchy is developed in the following figure (Figure 1.3).

The operating system keeps active programs and data areas in expanded storage. This is in addition to the most active programs and data areas, which are kept in central storage. A global LRU algorithm is used across the combination of central and expanded storage. The most active data are placed in central storage, the next most active data reside in expanded storage. Generally, items can remain unreferenced in expanded storage for hundreds or even thousands of seconds before they are moved out to a lower level of the hierarchy. The operating system accomplishes the movement to the level below expanded storage by first moving the data into central storage and then sending it to the DASD subsystem.

1.2.3 DASD subsystem

The base of the active storage hierarchy is the Direct-Access Storage Device (DASD). These devices are built on the magnetic disk technology with mechanical actuators, and they represent a highly evolved trade-off between storage cost and device performance.

The most critical measure of DASD performance is RESPONSE TIME, which is composed of two components: *service time* and *queuing time*.

A) SERVICE TIME

Service time is defined to be the sum of the following times: *seek*, *latency*, *Rotational Position Sensing* (RPS) *miss*, and *data transfer*. Service time is a raw measure of the performance of the DASD.

The time to move the actuator from one location on the disk to another is called *seek time*.

Latency is the time delay associated with the rotation of the disk storage medium until the requested data field is located under the read/write head. Generally latency is stated in terms of the time it takes to complete a half revolution of the disk.

Once the proper record on the disk has rotated under the read/write head, the device is ready to transmit data back to the channel. If the channel is busy servicing another device, the opportunity to transmit data is missed and a full rotation is required before the read/write head is properly positioned over the record again. This additional delay is called a *rotational position sensing* (RPS) *miss.* The number of misses that can be seen by a given I/O operation is a function of the utilization of the channel, control unit, and pathing configuration.

Data transfer time is the time it takes to move the data from the device to the central storage of the processor. It may be calculated by dividing the number of bytes to be transferred by the transfer rate of the DASD. This yields average transfer time in the range of 1 to 3 milliseconds.

With a seek time in the 4 to 5 millisecond range and latency above 8 milliseconds, data transfer is often not significant contributor to DASD performance. However, in batch, logging, and dump/restore applications, and cached-DASD subsystems, the seek and latency can be minimized, and transfer rate becomes dominant.

B) QUEUING TIME

Queuing time reflects the delay in initiating an I/O request, because the path to the device or the device itself is busy with another request.

Because of its effect on the overall system throughput and end-user, minimisation of DASD response time is a primary objective in the design of a storage hierarchy. Response time is minimized by reducing the impact of each of its components.

For instance, in the current DASD and control units, one of the more effective innovations is the use of alternate pathing to reduce path contention to a point that almost eliminates the RPS miss. In these systems the electronics provide up to four alternate paths from the device to central storage.

Significant contributors to DASD performance are based on mechanical rather than electronics technologies. Therefore, other avenues must be explored to keep up the pace with the DASD response time requirements. Data reference patterns, for one thing, lend themselves to incorporate a more intelligent management algorithm into the electronic storage technology used in a DASD control unit.

1.2.4 DASD cache control unit performance

The buffering of data in a high-speed, lookaside cache depends on a characteristic of data called locality of reference. In the case of sequential access to data, there is a probability that once a record is referenced, the succeeding record will also be accessed. If a request can be serviced from the cache, seek and latency times can be avoided. Thus overcoming the two most severe performance problems in the DASD subsystem.

Partial track staging provides two improvements. By staging the record referenced and the rest of the track following it, the cache anticipates sequential references and the nearby references associated with small random databases. With a little assistance from the LRU management of cache, this allows hit ratios in the high 9O percent range to be achieved, with reasonable cache sizes if the concerned datasets are appropriated to be cached. [NEA88]

A new generation of control unit came over with the fast write concept. The fast write capability provides a Non-Volatile Storage (NVS) in the controller that allows write hits to enjoy the same performance advantages granted to read hits to the cache. This way, seek and latency times are also avoided.

Together these features give a DASD cache controller up to an order of magnitude better service time and response time than an equivalent configuration of uncached DASD. The purpose of this section is not to discuss over the good cache candidate and the technical future environment of this new control unit generation.

Cache control unit and DASD provide comparable availability. Over time, DASD has become one of the most reliable components of a computing system. As a result, most recovery and availability software has made an assumption about the safety of data written to DASD. Whenever this assumption is wrong, the recovery is long, tedious, and costly. Thus the design of a cache control unit must offer availability and reliability characteristics very similar to those of the DASD.

The IBM 399O control unit structure and compatible provides a number of features that enhance availability as well as performance.
For some further explanation consult ANNEXE 1.A

1.2.5 Importance of the I/O boundary

The most distinct break between levels of the storage hierarchy occurs between the processor storage and the DASD subsystem levels. This point is referred to as the *I/O boundary* and denotes the place where data access switches from being synchronous with the processor, to being asynchronous.

During the execution of a task, a processor continually requests the next instruction or data item associated with the task, until the task is completed. The expected time to retrieve the requested item determines wether it is more efficient for the processor to wait for the requested item to arrive than to look for a different task to execute. The break-even point for this decision occurs when the time a processor sits idle waiting for an item to arrive equals the time a processor is busy switching among the tasks being executed. A retrieval is called SYNCHRONOUS when the processor waits for the item to arrive; a retrieval is called ASYNCHRONOUS when the processor looks for something else to do instead waiting.

An *asynchronous* retrieval is a disruptive event for both the hardware and the operating system. The operating system must take the following actions: set up for and schedule the retrieval; save the state of the original task; and search for a new task to dispatch. During the early stages of execution of a new task, the processor works slowly, as it waits for its HSB to be filled with data associated with the new task by displacing the HSB data of the original task. At some point in the future, the retrieval will be completed, which leads to a switch back to the original task and a repeat of the step just described.

The highest level of the DASD subsystem, the DASD under cache control unit or Solid State Device that simulate DASD, are capable of delivering data at cache speed.

Thus, levels of the storage hierarchy from expanded storage and above are accessed synchronously, levels from the DASD cache control unit and below are accessed asynchronously.

The most common implementation of an asynchronous retrieval is an I/O operation. To retrieve a data item in this manner, an application builds a request for an I/O operation. The operating system schedules the I/O and then performs a task switch. When the data are available, an I/O interrupt is processed followed by an eventual task switchback to the original task.

1.3 Importance of DASD within a storage hierarchy

1.3.1 Processor storage volatility

Volatility of storage means that the contents are lost when the power is shut off. Processor Storage (both central and expanded) is volatile. Therefore, all software has been designed so as to take this volatility into account. For example, I/O buffer managers and database systems force updates out of the buffer to the DASD subsystem before the updates are completed.

In a typical interactive database environment every updates must be written through to nonvolatile storage in the DASD subsystem.

1.3.2 Data placement within the storage hierarchy

Data placement within the storage hierarchy is automatically managed by the operating system, the subsytems, and the hardware. The systems programmers controls the eligibility of data to be retained at certain level. Essentially, all data are initially placed in the lowest levels of the hierarchy. Based on its eligibility status, the data may be moved into and held at the DASD cache control unit level and/or the processor storage levels. Entry into the highest level, processor high-speed buffer, is strictly a hardware function resulting from a processor's request to operate on a particular data item.

A data placement strategy is implemented through the behaviour of the data that are eligible to be retained in Processor Storage and those that may be retained in the DASD control unit cache. Data are made eligible for DASD control unit cache through placement within the DASD subsystem by the user. Data are made eligible for the Processor Storage by the operating system decision.

There are effectively no restrictions on the type of data that may reside in the DASD control unit. In addition, some DASD cache control unit have several features as DASD Fast Write, Dual Copy. The types of data that are attractive candidates for these DASD cache control unit include the following :

- data that are frequently READ and not appropriate to be held in processor storage,
- data with a user's need for frequent and/or fast update,
- data with high availability requirements,
- data being updated by more than one system.

1.4 Concluding remarks

A storage hierarchy is an image of the system structure to take best advantage of the total set of available storage technologies. It provides a solution to the key problems of representing storage performance and cost possibilities. Storage hierarchies have been successfully utilized in addressing the problems of a speed mismatch between the CPU and DASD, so called an I/O boundary. We establish that DASD behaviour is the most critical component in transactions.

The primary objective in the design and the management of a storage hierarchy, is the minimisation of the DASD response time which effect on the overall system throughput and end-user.

In the next chapter, we will look over the different available solutions we may use to measure and collect information concerning the Direct Access Storage Device behaviour.

2. Introduction to performance detection possibilities

System administrators are keeping busy tuning the storage to balance workloads and contention, and achieve acceptable performance.

Tuning is the study of the 'information-processing system' to use the available hardware at the maximum of its capabilities regarding the user requirements.

When we are talking over a computer system performance, we are talking over the performance *required*. The performance required depends on many factors, the weight given to this factor varies with the person and type of application. Different persons consider differently the same system because they have various requirements and aims.

Performance are represented by indexes. Each index can be an object of an *evaluation*, and can be evaluated in various ways.

It can be measured, calculated or estimated. If the results in performance values are inferior to the ones required, or if the possibility of improving the system's performance seems to exist anyway, a careful analysis of the causes of inefficiency and of the best remedies to it must be carried out. Performance evaluation tools will be used to perform this analysis.

2.1 Analysis procedure

Clearly, verification that the requirements are satisfied becomes more important with more critical performance requirements. The evolution of a computer systems performance is necessary not only during the system's productive life but also during its design or configuration modification. We will focus our interest on the improvement of a DASD subsystem efficiency. This improvement usually causes effects that are very important, but difficult to measure. We cannot hidden that the real purpose of these kinds of studies concern as much as the improvement of the "cost / performance", than the performance improvement itself.

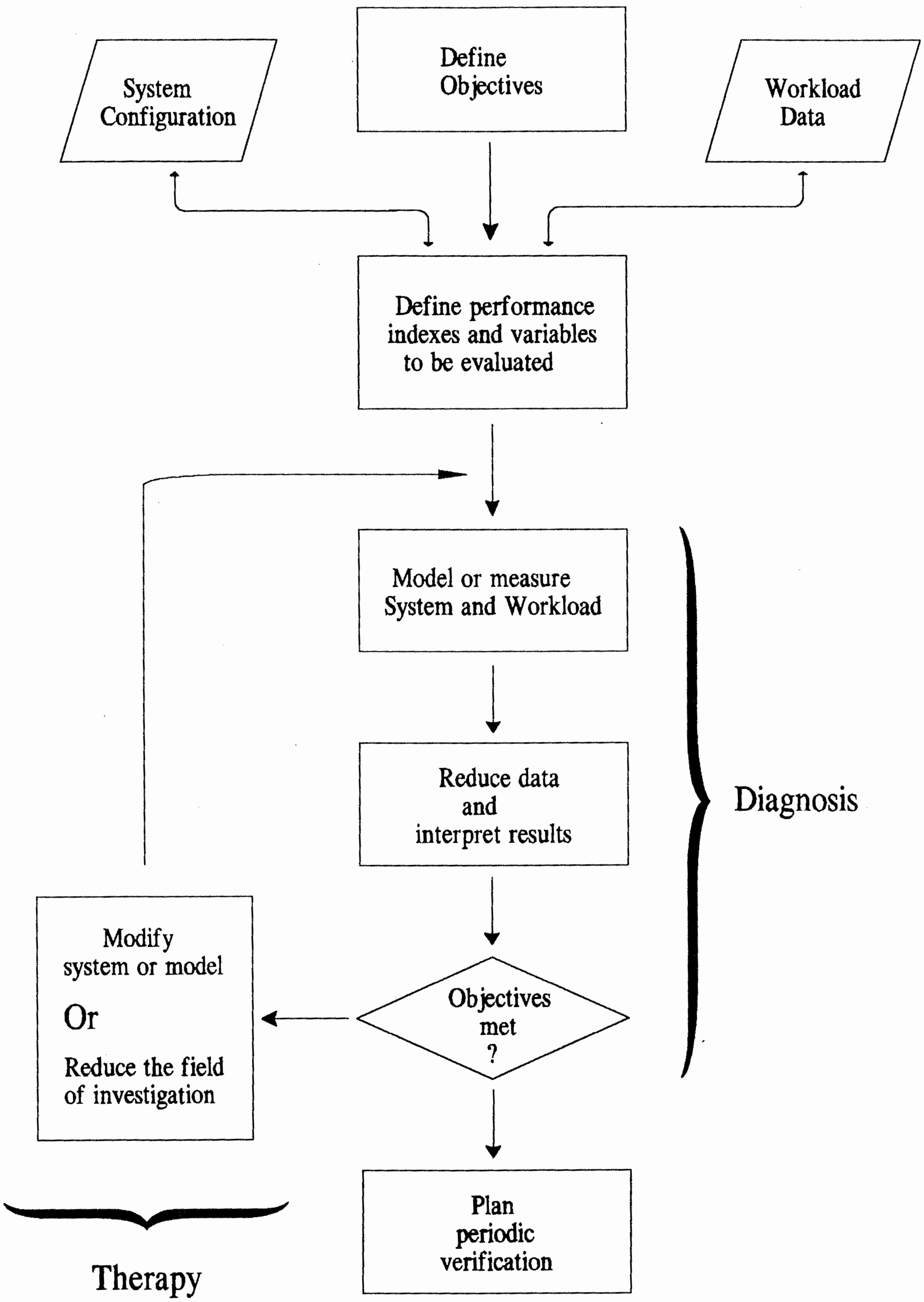
The typical procedure for attacking the problem of improving the "cost / performance" ratio is an iterative procedure. The cycling procedure contains two phases. The first one consists of a DIAGNOSIS phase and the second one of a THERAPY phase.

. The goal of the first phase is to diagnose the unsatisfactory performances.

. The second phase will be effective if it is capable of eliminating or "widening" the bottelneck that limit the subsystem performance or if it is capable to limit the field of investigation.

Obviously, the analyse in cost & time to perform the chosen therapy from a pool of therapy depends on many factors. We also have to state that the less expensive therapy should be to encourage prevention on problems that may occur.

Figure 2.1



System Configuration
Define Objectives
Workload Data
Define performance indexes and variables to be evaluated
Model or measure System and Workload
Reduce data and interpret results
Objectives met ?
Diagnosis
Modify system or model
Or
Reduce the field of investigation
Therapy
Plan periodic verification

The diagnosis therapy cycle outlined, or the limitation in regard of the field of investigation, must sometimes also be repeated because some other bottlenecks show up or because the problem is still too wide to diagnose.

We observe on the depicted following figure (`Figure 2.1`) that the workload of the studied system is also considered as a parameter of the method. The differences in load on the various components of a system caused by user program directly affect its global performance. One cannot evaluate a system without studying the characteristics of its workload. The "*workload* of a system" is made up of the programs, the data, and users requirements input to the system.

2.2 Performance indexes evaluation

We will now introduce the performance indexes, which are the primary quantities to be evaluated. There is various techniques to evaluate these quantities i.e. methods by which the indexes may be obtained.

2.2.1 Measurement categories

Measurements can be classified into two major categories, the measurement requested by system's analysts, and those requested by the system itself.

A) MEASUREMENTS REQUEST BY THE SYSTEM

Measurements requested by the system allow it to adapt dynamically to the factors conditioning its activity (mainly the workload).

B) MEASUREMENTS REQUEST BY THE PERFORMANCE ANALYST

> "In general, neither the hardware nor the software of a computer system is purposely designed to be measured". [FER83]
>
> This restricts the measurement that can be performed and if these restrictions needs to be overcome, some of the system functions have to be modified or special tools to be added.

To reach the objectives of a performance evaluation study, we must sometimes determine a *macroscopic* analysis, describing global indexes. When a higher degree of details are needed during a given time interval, the analysis will be *microscopic*.
What may distinguish these two types of analysis is the duration of the observed phenomena, the frequency these analysis are made, the tools they are using, and sometimes the significant overcost on the general system behaviour. What may also distinguish these two types of analysis is that stepping from the first one to the second one may be understood as a reduction of the field of investigation.

There are many techniques for measuring a computer system. The choice depends on the desired type of analysis and on the level of accuracy at which it has to be performed.

2.2.2 Event detection

An *event* is a change of the system's state. There is various way of collecting data about certain system activities.

> The first one is by *event detection* which means to capture all system's state variation and to record the associated indexes in the same order in which the corresponding event occurred.
>
> The second one is by *sampling*, this technique is often preferred to the one just described, because it does not interfere as much with the system processing. This technique means that at regular time intervals, the system is interrupted to detect the state of some of its components. This type of measurement may be sufficiently accurate if the number of samples collected are large enough.

Each performance measurement experiment requires a choice of the workload to be processed during the collection of the data. The workload itself may be natural or artificial and the type of workload does not influence the measurement technique, event detection or sampling, adopted to collect the data.
It is necessary to repeat the experiments, to perform a comparative evaluations.

2.3 Type of event

We have been describing various way of collecting data about system activities. We will now describe the different kind of *event* we may find in a computer environment.

A *software event* is said to be an event associated with a program's function.
A *hardware event* consists of the appearance of one or more signals in the circuits of a system component. We have to focus our attention on the fact that many hardware events can be recognized via software because they are accompanied by software event or environment modification.

The principle of software event detection is *to trap* the operating system. The operating system contains special code in specific places. When an event has to be intercepted, thé code is calling a special subroutine. The subroutine records the associated indexes of the event and stores them in a buffer area, and gives the hand back to the operating system.

A *monitor* is the set of instructions and data used for this purpose, therefore a monitor is a mechanism for collecting information on the system's activity.
The collected data will be processed at the end of the tracing period by an analysis program that performs a *data reduction* in order to make the data more easily interpretable.

We have just talked over the interception of event. A *full trace monitoring* may lead to the collection of an exorbitant mass of data, which will require a considerable amount of CPU time for their reduction. But this tracing technique provides an accurate information on certain aspects of a system's behaviour. We have to be aware that the use of such full tracing in software event measurements and some hardware event measurements must be selective since intercepting too many events would slow down the normal operation of the system to an unacceptable point. Some systems are equipped with hardware that makes event tracing less cumbersome.

The technique just described is based on the interception and recording of all the events of a given kind, which is applying the event detection technique. When the knowledge of the recorded items does not have to be so accurate or is to expensive to process, a *sampling* method may be adopted.
Sampling is a statistical technique. Instead of examining the whole set of events, the method gives images of the system's behaviour thanks to measurement intervals called samples. Based on the sample with a certain degree of accuracy, it is possible to estimate some values that are characteristics of the whole set of event. This technique presents the advantage of producing a much small amount of data, and will thus less slow down the normal behaviour of the system and shortening its analysis.

2.4 Sampling technique

We may use sampling technique for two different purpose.

The first one is *to describe* what happened during each interval with an a posteriori analysis. What's interesting to know for instance, is in what ratios the various events occurred, and how different types of activity were related to each other. The size of the sample determine the accuracy of the result. It should also be noted that the workload during each measurement interval has to be stationary to guarantee the validity of the results.

The second one is *to predict* the system's future behaviour; this way, we might be able to perform decisions that will have a positive interface on its performance. It is thus necessary in this case to create a model on an assumed underlying stochastic process able to simulate the system behaviour.

In some cases we must also take into account the workload variation when the studied system adapt itself dynamically.

Simulation models represent the dynamic behaviour of a system by reproducing its state and transition states. The transition states are caused by an appropriate sequence of external stimuli belonging to a natural or artificial workload. Based on simulation, we will measure and estimate the future behaviour of the studied system.

2.5 Concluding remarks

"The amount of data to be collected varies considerably as a function of the measurement technique and of the objectives of the investigation." [SAM90]

We emphasized the measurement, monitoring and simulation possibilities to evaluate a system behaviour and therefore the system inefficiencies to be identified.

After dealing with generalities about performance detection possibilities, we will take a look in the next chapter at the practical side of the matter, directly related to the training period at the 'ASLK-CGER Bank & Insurance'.

3. Performance problems in practice

After a basic description of the available products, a proposal made during the training period at the "ASLK-CGER Bank and Insurance" will be explained, and the existing situation seeking performance problems described. Some performance problems research will also be depicted as a sample of what may sometimes occur.

3.1 The choice between monitor and data collector

We will use the most current manner to classify the available performance measurement products. One frequently used division is between *monitors* and *data collectors*. Monitors present data as they measure it and data collectors store the data for future use. Sometimes both approaches are taken in the same measurement products, which means that some are immediately available while data representing past conditions are stored, also called data collectors in common langage.

As a primary technique to solve performance problems, we should do a choice between real-time monitors and data collectors. It is true that a small increase in response time for a specific device will be diagnose by a series of real time measurements, but a data collector with an appropriate reporter program can present the history of the device behaviour and can guide the analyst to a solution which is not unduly influenced by instantaneous peaks or events.

On the other hand, some particular problems must be revealed on-line and understood promptly. To fill this lack or this need, some real-time monitor includes additional specific function with a constantly active 'alert' capability. These monitors provides the best way to be aware of such critical problems.

The real-time monitors were primarily oriented to basic concerns affecting the *availability* of the hardware and the software configuration. [FER78]

A real-time performance monitor evaluates also service data and emits an alert if a critical response-time or elapsed-time service level has been passed over. It may also include a facility to decompose those times into their component parts, so that quickly we may identify the most likely cause of a service anomaly.

As developed and capable the real-time monitors may be, they are not the best choice for helping solving application performance problems.

"Such service anomalies tend to be chronic or cyclical, they do not wait until someone is looking at the display of a real-time monitor". [SAM90]

A data collector has advantages for performance management. Only the collector portion itself needs to run with a high priority level. The data analysis runs as a Batch job, and the display portion runs via interactive commands or programs.

This way of proceeding, makes possible a comparative evaluation of the collected measures and is performed afterwards.

3.2 The products in use

In an MVS environment, the System Management Facility (SMF) is used as a data repository.
A built-in data collection routines in the operating system gather the most basic measurements and reported through SMF. Reporting must be deferred until the SMF data is dumped and processed.

There are some benefits of routing performance measurement data through SMF. The 'process' is divided into two parts, the *basic measurement* and the *data collector*. As the data collector is deferred, there is less overhead attributable to it. The facilities provided with the operating system and designed for continuous operation write automatically the data out.

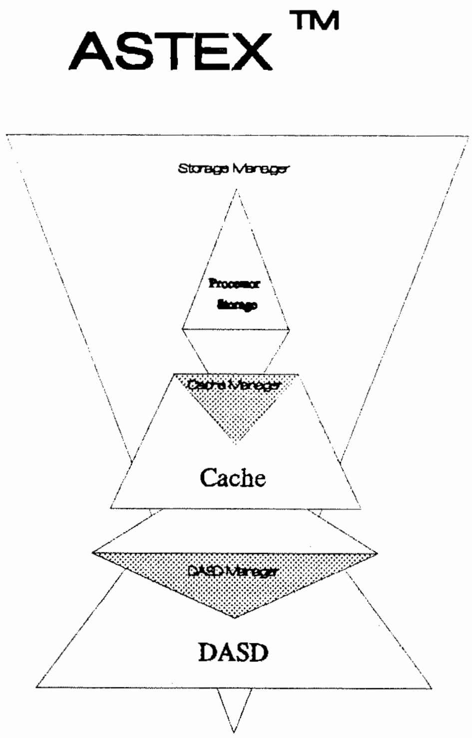
There are also several complementary products available for data analysis. Those includes IBM's Service Level Reporter and SAS Institute's Statistical Analysis System.
Higher-level data base and reporting packages built on SAS has been developed, including Legent's MICS product.

Another basis for classifying measurement products is according to wether data is obtained by sampling, by direct access to existing summary information, or by intercepting events.

3.2.1 Resource Measurement Facility

IBM's Resource Measurement Facility (RMF) is primarily a sampling system. We know that basic measurement data at the workload level are available from SMF, but it is also necessary to have an overview of how the system is performing as well.
The relationship between system capacity and workload is made possible using data collected by RMF and written to SMF datasets in the form of specific records.

Figure 3.1


ASTEX ™
Storage Manager
Processor
Storage
Cache Manager
Cache
DASD Manager
DASD

3.2.2 Generalized Trace Facility

The Generalized Trace Facility (GTF) is an event detection monitor. The use of GTF is considered abnormal; most installations require special authorization to allow its use, and only for limited time intervals due to the provoked CPU overhead. MVS includes a 'hidden' set of data collection facilities used by GTF.

3.2.3 ASTEX from Legent

The ASTEX product from Legent has been developed to automate the performance process at multiple levels of the storage hierarchy. ASTEX is made of different components which consists of the Storage Manager, Cache Manager and DASD Manager. This three components work together to optimize the utilization of specific cache and DASD resources as well as managing service level objectives, and recommending solutions to response time problems across the DASD concerned part within the storage hierarchy. The structure of the ASTEX product can be viewed in the following figure (Figure 3.1).

Building a complete picture of performance, the *Storage Manager* suggests solutions to response time problems for specific workloads. Solutions may be simulated to eliminate new problems from occurring.

The *Cache Manager* analyzes cache usage, provides recommendations for actions that will maximize the performance benefits of cache and Non-Volatile Storage (NVS).

The *DASD Manager* provides the data needed to solve specific DASD problems with organized online displays and batch reports, and may also be used to insure efficient DASD utilization.

3.3 Personal approach of performance problems

We explained in the first chapter that the minimisation of DASD response time is a primary objective in the design and the update of a storage hierarchy, and have a major effect on the overall system throughput and end-user. We described in the second chapter at a theorical point of view the different measurement possibilities available and afterwards the corresponding practical tools available.

3.3.1 Personal product description

The next following discussion will introduce and develop the proposal made during the training period at the "ALSK-CGER Bank and Insurance".

In designing and observing the performance at a DASD level, each technologies have to be used at the maximum of their capabilities. To do this at a practical way, each device has to be considered has a unique case, based on the kind of information the device has to support and the job it has to achieve.

The major problems occurring using DASD are of two kinds, either physical or logical. Physical problems are most of the time directly related with the state of the concerned device, and logical problems are related to the manner the device is used. Both types may be sometimes confuse.

Sometimes after applying a couple of therapy, it is not possible any more to improve the performance of devices even if the expected performance is not achieved. Diagnosis phase is currently based over specified thresholds bypassing.

Supposing each set of devices identifies by a set name. Supposing also, as a problem detection basis, each set name is associated with a threshold.
If the measured set behaviour is greater than the specified threshold associated with the set, this will be interpreted as a detected problem. Otherwise it will be interpreted as a non existing problem.
If one device only from a set has a behaviour such as a threshold bypassing occurred. The set name associated with a such device will show up in a performance list. This kind of list is called an exception list.
As an interpretation of this exception list, we cannot say the entire set is causing a performance problem.
This is the mainly reason why, as a problem detection basis, each device has to be associated with a threshold.

Considering we apply a general threshold for each device; the constructed exception list, always based over threshold bypassing, should only contain 'true' problems.

Sometimes the pointed problems are real, sometimes they cannot be solved, sometimes they are just the consequence of workload peaks. In either case, to insure that the exception list are full of 'true' problems, we have to consider each device particularly ***with its own threshold***.
The threshold should not be associated to a set of devices any more but to each device particularly.

Just taking a look over the latest suppositions, considering performance behaviours that cannot be improved any more for any reasons. While thresholds are associated with the devices themselves, a particular threshold may be emitted to this kind of device, and they won't appear any more in the exception list.

We just thus introduced the difference between 'true' and 'false' problems.

Sometimes device thresholds bypassing are just the consequence of well known workload peaks. If these workload peaks are periodic, we may consider the detected thresholds bypassing as 'false' problems. To eject these 'false' problems from the exception lists, we should apply a specific threshold associated with the period of the detected workload peaks.
For instance, some devices are supporting the interactive environment, or data base working with specific applications, either we want to specify an higher threshold for periodic peaks, or we don't want to hear about the bypassed threshold even as bad the performance might be.
Special thresholds may be enabled only for a certain period of time (a few quarter fo instance). This kind of thresholds establishment allows a very accurate setting of the performance problems research.

We are usually associating thresholds with the concept of no response time bypassing. In this case the concept of threshold will be also associated with any reported type of indexes; for example, any of the response time components.

In practice, the developed tool has to be progressively configured to diagnose 'true' problems. The tool may also be used as a workload analyser while it gives the possibility to the analyst to carry out a all set of measures for a given period. This tool is working at the macroscopic level of the diagnosis phase. Afterwards the performance analyst will be able to compare, and classify all the measured devices behaviour according to his own particular research plan and method.

3.3.2 Description of the existing situation

We have just explained the philosophy of an 'automated' performance problems detector based on the concept of 'true' and 'false' problem. Due to the increasing amount of storage capacities, the need of such a tool is obvious. But we haven't yet described the existing situation where this study has been made. Basically, there was a major pressure to treat and detect problem at a time interval level for each device.

In practice for performance bulk studies, the day is divided into two periods. The first period is associated with day time and the second period with night time. An exceptional third period is created concerning the week-end and the closing business day.

The storage managers focus their interests over the devices behaviour during day-period-one. The used technique consists to split the demarche into two steps.
The first one consist to create an exception list based on a further explained calculation method, and the second one to look in details over some particular devices behaviour via the lately invested Legent software. (ASTEX)

We will now describe the contents of the exception list made during the first step.
First of all, a calculated response time is associated with each device every day-period-one.
If this calculated response time is greater than one unique predefined threshold for ***all*** the devices, it will be considered as a threshold bypassing and will be included into the exception list.

This response time calculation is quiet complex. Its calculation is based on the response time evaluation available via RMF. Each response time measure available represents a device behaviour during a quarter. The response time needed for the calculation are those included in day-period-one.

The calculated response time equals to the summation of these measured response time, if they are respecting a particular feature, divided by the number of measured response time taking in account for the summation. The particular feature consists to say that each measured response time may be used for the summation if its associated activity rate is greater than 2 .

$$\text{Calculated Response Time} = \frac{\sum_{i=1}^{T} \text{Response Time}_i * £_i}{N}$$

T = Number of quarter in day-period-one

$£_i = 1$ if activity rate > 2 else $£_i = 0$

N = the number of Response Time respecting the feature

$$N = \sum_{i=1}^{T} £_i$$

The depicted formula is obviously not performed if the number of Response Time respecting the feature is equal to zero.

3.3.3 Critics of the existing situation

Some of the arguments developed during the '3.3.1 Personal product description' section may be turned into critical arguments for the existing situation. Some other critics may also be depicted, such as :

. No study of the measure has never been performed at a day-period-two level. While the Batch job is not affecting the end-user, it affects the overall system throughput !

. All the type of storage concerned, which includes Solid State Devices, cached and non-cached DASD are treated exactly the same way, by the use of an unique threshold !

. Lots of problems are bypassed because they are lost into the average response time calculation. Sometimes also bad performance devices are showing up in the exception lists but they are just workload peaks image.

The major facilities of this kind of exception list is in a certain manner to reduce the field of investigation. Meanwhile, some recently improvements have been made concerning the construct of the exception lists. Such as cached and non-cached devices are not reporting on the same list any more.

An example of the described report is available in Annexe 3.A.

3.3.4 Personal tool process description

The tool process is based over SMF[1] and RMF[2] measurements and records in a first part, and over the MICS reduction report in a second part. The MICS report is an input set of raw data for the developed tool. Once the storage manager started describing some thresholds, the tool execution may begin being effective.

While no thresholds exist, any difference can be made between 'false' and 'true' problems. Otherwise, the differences are made by comparing the defined thresholds and the devices behaviour image given by the MICS reports. The 'true' problems are depicted by alert signs applied during the comparison.

In a last step a user interface is provided to help the storage manager in his research.

We will now talk over the tool process itself. The tool process is cutted in three phases, which may be defined as Batch process, Link between Batch and interactive, and the interactive process.

A) BATCH PROCESS

The Batch process is initialized by a user requirement. The user specifies the system, the day-period (one or two) and the day which is concerned by the execution. A squeleton procedure will be turned into an executable procedure based on the introduced parameters. In a last step, the executable procedure is submitted.

---

[1] SMF : System Management Facility

[2] RMF : Resource Measurement Facility

B) LINK PROCESS

If the Batch process executed correctly, the link process will be enabled. The link process consists to work with a mailbox dataset. Every time a batch process ends correctly, some parameters concerning the Batch execution will be written in the mailbox dataset.

C) INTERACTIVE PROCESS

The interactive process via the user-interface has different functions added to the principal one, which is to help the storage administrator within his performance problems research.

- The first one is to create and update progressively the set of thresholds.
- The second one consists to submit the Batch process.
- The last one consists to compare the defined set of thresholds and the devices behaviour given by the MICS reports. MICS reports are made available to the interactive process by the information included within the mailbox dataset.

A detail tool process description with a copy of the panels interface are available in the Annexe 3B

3.3.5 Personal tool critics

The developed tool has been made to fill a lack in regard of the performance problems research.

This tool will be very constrained during the first part of its implementation while the thresholds have to be defined.

This tool doesn't have the pretension to detect all the performance problems research, but just another manner to look at them and to work in complementary with the other packages.

In the current situation, the developed tool is still at a prototype stage because some problems occurred running it with the production data environment.
The time needs to perform the Batch process is huge due to the MICS & SAS interface. Even if the Batch process may be introduced in the Planning for user accommodation, it doesn't reduce the needed execution period.

Keeping the idea, because it showed to be *useful* by the performance analysts, another solution is on its way and has already been implemented by the performance analyst team.
It consists to grab the measures directly in the RMF records without the MICS & SAS interface.

3.3.6 Tool situation among others

The described tool is of course working at the macroscopic level of the diagnosis phase. If a deeper research concerning detected problems has to be performed, we will use other packages at a microscopic level this time. We might use the ASTEX performance database to look for instance after devices behaviour or job name concerned by a bad performance measure, the GTF[3] tracing system, or configuration reporting tools.

The former detection performance problems method enabled by the explained exception list and its response time calculation is still available.
This exception list is still used as a departure in the problem detection environment. Just as other reports may also be used.

It's obvious that in most cases, the performance analyst will start his research from the most general collected set of information, to the most depicted set of information. As some rules of thumb exist over the *interpretation* of performance problems, the research method suggested in section 2.1 'Analysis procedure' is implicitly applied by the analyst.
Implicitly because each research based its plan over the analyst and the storage administrator knowledges, and the particular devices envolve. For instance, they might diagnose immediatly the exact cause(s) of performance problem without following any general methods.

The collaboration between the performance analyst, the storage administrator and the application manager is a prime necessity, and turn to be merely an organisation problem than a 'computer' problem.

[3] GTF : General ized Trace Facility

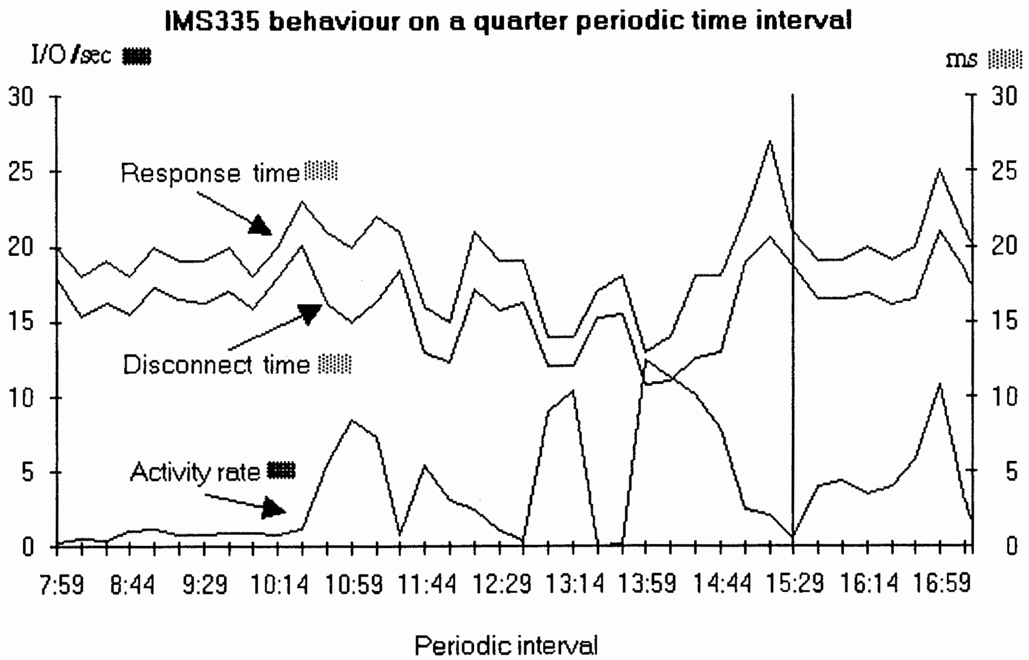
In practice, it's utopic to suggest a general method that will never be followed, but better suggesting a third hand tool given the performance analyst another manner to watch his working environment.

Two examples based on real case performance studies will be depicted in the following section. By these two examples, the heaviness of the current used demarch and the need of an automated I/O management will be illustrated.

3.3.7 Example I

The first example is based on a repeated threshold bypassing from both the response time and the disconnect time by a specific device (IMS335).

The following figure depicts this device behaviour during a day-period-one. By this figure, we can establish that an increase in response time is directly related to an increase in disconnect time.



Via the ASTEX online performance checking system, we suggest on the next figure an average Response Time analysis panel depicting the 15:15 - 15:30 interval.

Summary: Group View : Storage Interval 15:15 - 15:30 ROW 20 OF 99
Detail : Volume Screen: Response Analysis As of 15:27 on SYSA 04 APR 91
) Scroll CSR

Volume	DUA	I/O %	Obj Time	Obj %	Mode	Prob% Brp%	Prob% Vol%	Rsp AVG	Rsp Components IOSQ	Pend	Conn	Disc	Dsp
a	m	c	n	o	p	e	f	g	s	t	u	v	w
IMDB		100.0			E	2.9		17	1.3	0.3	2.2	12.8	0.2
IMS4CD	8E2	0.0	50	90	E	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0
IMS338 +	89E	2.4	50	90	E	0.0	0.0	14	0.1	0.3	3.9	9.5	0.1
IMS337 +	88E	1.4	50	90	E	0.0	0.8	17	0.8	0.3	2.7	12.7	0.1
IMS336 *	86E	8.6	50	90	E	1.1	13.1	28	5.1	0.2	2.3	20.0	0.2
IMS335 +	86C	2.7	50	90	E	0.1	5.6	19	1.9	0.2	2.3	14.8	0.1
IMS334 +	86B	2.5	50	90	E	0.1	3.2	12	1.8	0.2	2.0	8.3	0.2
IMS333 +	86A	1.3	50	90	E	0.0	1.6	11	0.1	0.2	2.0	8.4	0.2
IMS332 +	869	1.2	50	90	E	0.0	1.7	12	0.9	0.2	2.1	8.9	0.1
IMS331	867	0.0	50	90	E	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0
IMS330	866	0.0	50	90	E	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0
IMS329 +	865	4.0	50	90	E	0.0	1.2	6	0.6	0.2	1.9	2.8	0.1
IMS328	864	0.0	50	90	E	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0
IMS327	863	0.0	50	90	E	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0
IMS326 +	877	1.3	50	90	E	0.0	1.0	11	0.5	0.2	2.3	7.6	0.1
IMS325 +	889	1.0	50	90	E	0.0	1.0	20	0.6	0.3	3.1	15.4	0.2
IMS324 +	898	0.3	50	90	E	0.0	0.0	23	0.0	0.3	2.5	19.7	0.1
IMS323 +	892	4.1	50	90	E	0.1	2.4	13	2.0	0.3	2.5	8.5	0.1

At that current time, we establish that the average response time is equal to 19 milliseconds with a disconnect time of 14.8 milliseconds.
It might be interesting to look at a dataset level to find out which datasets are invoking a so big disconnect time.

The following figure demonstrates an average Response Time analysis at a dataset level.

Summary: Volume View : Storage Interval 15:15 - 15:30 ROW 1 OF 3
Detail : Dataset Screen: Problem Analysis As of 15:27 on SYSA 04 APR 91
) Scroll CSR

Dataset	Total I/O	I/O %	I/O SEC	Problem Type	Prob% Lst%	Prob% Dsn%	Problem Distribution Pth%	Rsv%	Vol%	Sk%	Dsp%	Dsn%
a	b	c	d		e	f	g	h	i	j	f2	k
IMS335	2,065	100.0	2.8		9.9		46	0	17	10	0	27
I5.LR.DA.LR310201.CD	975	47.2	1.3		4.8	10.3	70	0	10	0	0	20
I5.TT.DA.TTDOOP01.CD	845	40.9	1.1		3.6	8.9	27	0	20	7	0	47
I5.LR.DA.LR310101.CD	245	11.9	0.3		1.5	12.2	17	0	33	50	0	0

******************************* BOTTOM OF DATA ********************************

The depicted datasets are the only one in use at the monitored time.

We will show in the following figures these datasets behaviour components during the studied interval.

It might be interesting to look at a dataset behaviour components level, to find out which dataset is causing the performance trouble.

```
Summary: Dataset     View  : Storage          Interval : 15:15 - 15:30
                     Screen: Data Set Analysis  As of 15:27 on SYSA   04 APR 91
)
Dataset       Total  % Dsn   —Dsn Problem Distribution——     Dsorg   Last Job
Volume         I/O   Probs   Intra Seek%   DsnQ%    Other%
....a.........b......a2.........b2.........c2........d2......e2........y1....
I5.LR.DA.LR310201.CD
IMS335         975     2         0          100        0      VSAM    LRU2001

 Program Managment Components:             Intra Seek Components:

  % of Total Dsn I/O         =        0     Avg Intra Seek Distance  =     130
  Bldl/Find %                =        0     Intra Seek % of Dsn I/O  =    55.9
  Fetch %                    =        0     No. of Allocated Extents =       8

 IOS Queue Components:                     Other Components:

  Avg IOS Queue time         =      2.5     Avg Latency+Connect Time =     9.5
                                            Avg Latency Time         =     7.2
                                            Avg Connect Time         =     2.4
```

```
Summary: Dataset     View  : Storage          Interval : 15:15 - 15:30
                     Screen: Data Set Analysis  As of 15:27 on SYSA   04 APR 91
)
Dataset       Total  % Dsn   —Dsn Problem Distribution——     Dsorg   Last Job
Volume         I/O   Probs   Intra Seek%   DsnQ%    Other%
....a.........b......a2.........b2.........c2........d2......e2........y1....
I5.TT.DA.TTDOOP01.CD
IMS335         845     4         0          100        0      VSAM    TTXEDEB

 Program Managment Components:             Intra Seek Components:

  % of Total Dsn I/O         =        0     Avg Intra Seek Distance  =     323
  Bldl/Find %                =        0     Intra Seek % of Dsn I/O  =     8.3
  Fetch %                    =        0     No. of Allocated Extents =      15

 IOS Queue Components:                     Other Components:

  Avg IOS Queue time         =      3.6     Avg Latency+Connect Time =    10.0
                                            Avg Latency Time         =     7.7
                                            Avg Connect Time         =     2.4
```

The 'Intra Seek Components' which in the described cases relatively high. Intra-seek problems often occur when a data set occupies several non-adjacent extents on a DASD. The random access of large datasets also causes intra-seek problems. To insure the thesis of the datasets fragmentation, we will use a special configuration disk analyser tool.

The following figure presents a list depicting the disk extents sequentially with their supported datasets.

```
INPUT:  Volume serial ===) IMS335
        Sort type     ===) NAME           specify NAME, EXTENT, DATE, or EDATE


MVS/DITTO                                        4/04/91 (91-094) 15:32  PAGE    1
* * * * DEVICE  86C  VOLSER= IMS335,                                        * * * *
* * 3380, WITH 1770 CYLS, 15 TRKS/CYL, AND 47968 BYTES/TRK                        *
---- FILE NAME ---- (SORTED BY EXTENT) ------ EXTENT  BEGIN-END      RELTRK,
1...5...10...15...20...25...30...35...40.... SEQ CCC-HH   CCC-HH     NUMTRKS

SYS1.VTOCIX.IMS335                            0    0  1     0 14   1,14
 *** VTOC EXTENT ***                          0    1  0     1 14   15,15
SYS1.VVDS.VIMS335                             0    2  0     2  9   30,10
I5.BK.DA.BKFOUT01.CI                          0    2 10     2 10   40,1
***
I5.BK.DA.BKKOKD01.CI                          0    2 11     2 11   41,1
I5.BK.DA.BKMVEN01.CI                          0    2 12     2 12   42,1
I5.BK.DA.BKTVEN01.CI                          0    2 13     2 13   43,1
I5.BK.DA.BKUITS01.CI                          0    2 14     2 14   44,1
I5.LR.DA.LR310101.CD                          0    3  0    57 14   45,825
I5.LR.DA.LR310101.CD                          1   58  0   112 14   870,825
I5.LR.DA.LR310101.CD                          2  113  0   167 14   1695,825
I5.LR.DA.LR310101.CD                          3  168  0   222 14   2520,825
I5.LR.DA.LR310101.CD                          4  223  0   277 14   3345,825
I5.TT.DA.TTDOOP01.CD                          0  278  0   332 14   4170,825
I5.FC.DA.FCSTOCX1.CD                          0  333  0   387 14   4995,825
I5.BK.DA.BKFOUT01.CD                          0  388  0   388 14   5820,15
I5.BK.DA.BKKOKO01.CD                          0  389  0   389 14   5835,15
I5.BK.DA.BKMVEN01.CD                          0  390  0   444 14   5850,825
I5.BK.DA.BKTVEN01.CD                          0  445  0   499 14   6675,825
I5.BK.DA.BKUITS01.CD                          0  500  0   500 14   7500,15
I5.BK.DA.BKVGKO01.CD                          0  501  0   501 14   7515,15
I5.BK.DA.BKVGKO01.CI                          0  502  0   502  0   7530,1
  *** FREE EXTENT ***                         0  502  1   502 14   7531,14
I5.BK.DA.BKMVEN01.CD                          1  503  0   557 14   7545,825
I5.BK.DA.BKMVEN01.CD                          2  558  0   612 14   8370,825
I5.FC.DA.FCSTOCX1.CI                          0  613  0   613 14   9195,15
I5.TT.DA.TTDOOP01.CD                          1  614  0   668 14   9210,825
***
I5.TT.DA.TTDOOP01.CD                          2  669  0   723 14   10035,825
I5.TT.DA.TTDOOP01.CD                          3  724  0   778 14   10860,825
I5.TT.DA.TTDOOP01.CD                          4  779  0   833 14   11685,825
I5.TT.DA.TTDOOP01.CD                          5  834  0   888 14   12510,825
I5.TT.DA.TTDOOP01.CD                          6  889  0   943 14   13335,825
I5.TT.DA.TTDOOP01.CD                          7  944  0   998 14   14160,825
I5.TT.DA.TTDOOP01.CD                          8  999  0  1053 14   14985,825
I5.TT.DA.TTDOOP01.CD                          9 1054  0  1108 14   15810,825
I5.FC.DA.FCSTOCX1.CD                          1 1109  0  1163 14   16635,825
I5.TT.DA.TTDOOP01.CD                         10 1164  0  1218 14   17460,825
I5.LR.DA.LR310201.CD                          0 1219  0  1328 14   18285,1650
I5.LR.DA.LR310201.CD                          1 1329  0  1438 14   19935,1650
I5.LR.DA.LR310201.CD                          2 1439  0  1548 14   21585,1650
I5.LR.DA.LR310201.CD                          3 1549  0  1658 14   23235,1650
I5.TT.DA.TTDOOP01.CD                         11 1659  0  1713 14   24885,825
  *** FREE EXTENT ***                         0 1714  0  1767 14   25710,810
I5.BK.DA.BKMVEN01.CD                          3 1768  0  1822 14   26520,825
I5.LR.DA.LR310201.CD                          4 1823  0  1932 14   27345,1650
I5.LR.DA.LR310201.CD                          5 1933  0  2042 14   28995,1650
I5.TT.DA.TTDOOP01.CD                         12 2043  0  2097 14   30645,825
I5.TT.DA.TTDOOP01.CD                         13 2098  0  2152 14   31470,825
I5.LR.DA.LR310201.CD                          6 2153  0  2262 14   32295,1650
I5.LR.DA.LR310201.CD                          7 2263  0  2372 14   33945,1650
***
I5.TT.DA.TTDOOP01.CD                         14 2373  0  2427 14   35595,825
MVS/DITTO                                   4/04/91 (91-094) 15:32  PAGE    2
THIS VOLUME IS CURRENTLY  96 PER CENT FULL WITH   824 TRACKS AVAILABLE
***
```

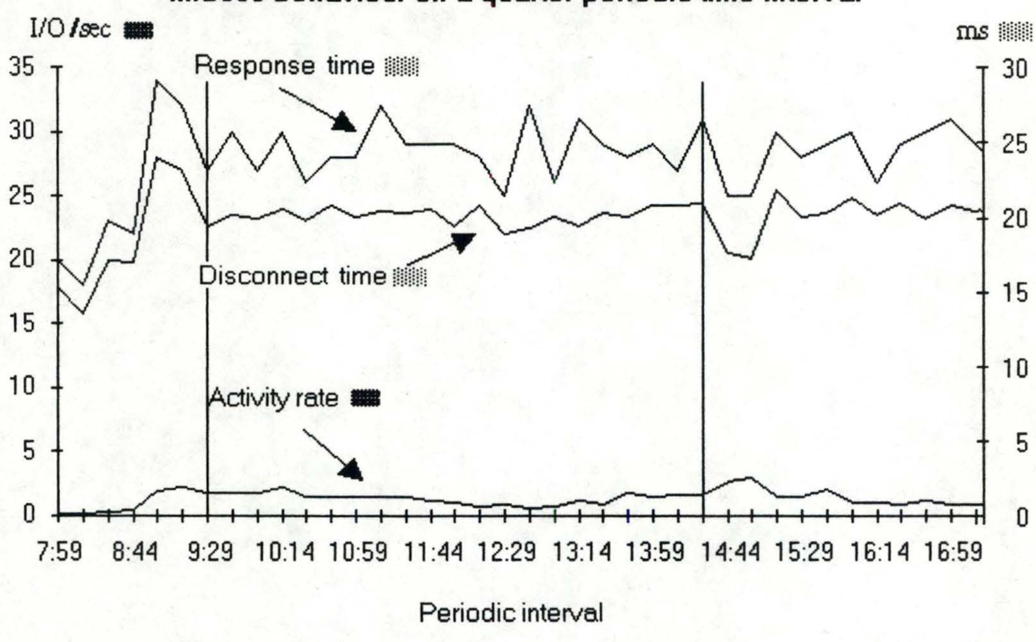
To reduce intra-seek problems, we have to be sure that dataset are made of one extent. The use of reorganization utilities to reorganize the contents of both partioned and non-partioned datasets is a must in this case.

3.3.8 Example II

The second example is also based over a repeated threshold bypassing from both the response time and the disconnect time by a specific device (IMS306).

The following figure depicts this device behavior during a day-period-one. We can establish that the response time is mainly so high due to its disconnect time component.

IMS306 behaviour on a quarter periodic time interval



Via the ASTEX performance data base, we suggest on the next figure a Response Time analysis panel. We chose arbitrarily the 14:15 - 14:30 interval.

```
            I/O      --Prob%---    Rsp    --------Rsp Components-------       Last
 Dataset     %       Lst%  Dsn%    AVG    IOSQ  Pend  Conn  Disc   Dsp       Access
.zl..a......c......e.....f.......g......s.....t.....u.....v.....w........nl......
 IMS306     100.0   10.3            30     4.7   0.2   2.3  22.9   0.1
    I5.ES.DA.ESINDX06.CD
             27.8    0.0    0.0     23     0.0   0.1   2.2  20.5   0.0     14:29:59
    I5.ES.DA.ESMRMI07.CD
             17.0    4.1   24.2     38    10.0   0.3   2.4  25.5   0.0     14:29:59
    I5.ES.DA.ESSPRK09.CD
             13.4    2.6   19.2     35     9.9   0.2   2.3  22.1   0.3     14:29:59
    I5.ES.DA.ESSPRK15.CD
             13.4    3.1   23.1     37     9.5   0.2   2.3  24.5   0.0     14:29:59
    I5.ES.DA.ESSPRK16.CD
             11.9    0.0    0.0     28     1.0   0.2   2.3  24.6   0.3     14:29:59
    I5.ES.DA.ESSPRK06.CD
             11.9    0.5    4.3     26     1.1   0.2   2.2  22.1   0.1     14:29:59
    I5.ES.DA.ESSPRK14.CD
              2.6    0.0    0.0     28     1.6   0.2   2.3  23.9   0.0     14:29:59
    I5.MO.DA.MOCELLI1.CI
 DFSXDSP0     1.5    0.0    0.0     30     5.9   0.3   1.7  22.5   0.0     14:29:59
    I5.MO.DA.MOCELLI1.CD
```

At that current interval, we establish that the average response time for the device itself is equal to 30 milliseconds with a disconnect time of 22,9 milliseconds. The datasets listed are participating in the device behaviour in front of their respecting activity rate as suggested in the column c.

The next figure will present a Seek Analysis concerning these datasets.

```
Summary: Volume  View  : Storage         Interval  14:15 - 14:30      ROW 1 OF 9
Detail : Dataset Screen: Seek Analysis       As of 14:30 on SYSA     18 APR 91
)                                                                   Scroll CSR
          Total    I/O     ---Total Seek---  ---Inter Seek---  ---Intra Seek---
 Dataset   I/O      %       AVG  Lst%  Dsn%   AVG  Lst%  Dsn%   AVG  Lst%  Dsn%
.....a......b......c.......o1....p1....q1...r1....s1....t1....u1....v1....w1.
 IMS306      970 100.0      928  92.8        1,092  77.8          74  14.9
    I5.ES.DA.ESINDX06.CD
             270  27.8      970  26.8  96.3 1,292  20.1  72.2      2   6.7  24.1
    I5.ES.DA.ESMRMI07.CD
             165  17.0    1,035  15.5  90.9 1,271  12.4  72.7     94   3.1  18.2
    I5.ES.DA.ESSPRK09.CD
             130  13.4      813  11.9  88.5   877  10.8  80.8    142   1.0   7.7
    I5.ES.DA.ESSPRK15.CD
             130  13.4      699  12.9  96.2   845  10.3  76.9    116   2.6  19.2
    I5.ES.DA.ESSPRK16.CD
             115  11.9    1,208  10.8  91.3 1,208  10.8  91.3      0   0.0   0.0
    I5.ES.DA.ESSPRK06.CD
             115  11.9      809  10.8  91.3   905   9.3  78.3    232   1.5  13.0
    I5.ES.DA.ESSPRK14.CD
              25   2.6      882   2.6 100.0   882   2.6 100.0      0   0.0   0.0
    I5.MO.DA.MOCELLI1.CI
              15   1.5      882   1.5 100.0   882   1.5 100.0      0   0.0   0.0
    I5.MO.DA.MOCELLI1.CD
```

The 'Average Inter-Seek' valueshows the average distance the DASD arm traveled while doing seeks for a series of different datasets.

The 'Average Intra-Seek' value shows the average distance the DASD arm traveled for a single dataset I/O request.

The high participation rate of the disconnect time within the response time may come from the need of specific datasets for a single application.
This conclusion is based on the dataset's name depicted on the presented panel. From the application knowledges, the application reads more less at the same time the pointed datasets.

To ensure this diagnosis, the following figure suggest another Seek Analysis panel for another arbitrarily chosen interval. The suggested one is 9:15 - 9:30.

```
Summary: Volume  View  : Storage          Interval  09:15 - 09:30    ROW 1 OF 20
Detail  : Dataset Screen: Seek Analysis        As of 09:30 on SYSA   18 APR 91
)                                                                    Scroll CSR
         Total    I/O      ---Total Seek---   ---Inter Seek---   ---Intra Seek---
  Dataset  I/O     %        AVG   Lst%  Dsn%  AVG   Lst%  Dsn%  AVG   Lst%  Dsn%
.....a......b......c.......o1....p1....q1....r1....s1....t1....u1....v1....w1.
  IMS306  13,141 100.0      130   85.9         195   56.1           8   29.8
      I5.MO.DA.MODIBU01.CD
             6,310  48.0     71   38.9  82.2   122   22.4  47.4     2   16.4  34.8
      I5.MO.DA.MOPERS01.CD
             5,446  41.4     90   38.0  91.1   130   24.8  59.5    15   13.2  31.6
      I5.MO.DA.MODIBUX1.CD
               270   2.1     61    1.7  83.3    61    1.7  83.3     0    0.0   0.0
      I5.MO.DA.MODIBUX2.CD
               240   1.8     97    1.6  87.5    97    1.6  87.5     0    0.0   0.0
      I5.ES.DA.ESINDX06.CD
               181   1.4    434    1.4 100.0   472    1.3  91.9     1    0.1   8.1
      I5.ES.DA.ESMRMI07.CD
               152   1.2  1,743    1.0  87.5 1,805    1.0  84.4    78    0.0   3.1
      I5.MO.DA.MOCELL01.CD
               110   0.8    167    0.4  50.0   167    0.4  50.0     0    0.0   0.0
      I5.MO.DA.MOCELLX1.CD
                95   0.7     13    0.3  47.4    13    0.3  47.4     0    0.0   0.0
      I5.ES.DA.ESSPRK16.CD
```

Based on the dataset's names depicted on the presented panel, we can conclude we are observing a similar phenomena as the first one.

This disconnect time problem comes from application needs. There is two ways to solve this problem.
- The first one consists to split the datasets from the same application on different disk.
- The second one consists to group physically the datasets from the same application on the disk to reduce th average Inter-Seek.

The following figure presents a list depicting the disk extents sequentially with their supported dataset.

```
INPUT:  Volume serial ===) IMS306
        Sort type     ===) NAME           specify NAME, EXTENT, DATE, or EDATE


MVS/DITTO                                         4/23/91 (91-113) 12:06  PAGE    1
* * * * DEVICE  87E  VOLSER= IMS306,                                        * * * *
* * 3380, WITH 1770 CYLS, 15 TRKS/CYL, AND 47968 BYTES/TRK                        *
---- FILE NAME ---- (SORTED BY EXTENT) ----- EXTENT  BEGIN-END     RELTRK,
1...5...10...15...20...25...30...35...40.... SEQ CCC-HH    CCC-HH    NUMTRKS

I5.MO.DA.MOFETSI1.CI                          0     0  1     0  1   1,1
I5.MO.DA.MOHSLDI1.CI                          0     0  2     0  2   2,1
I5.MO.DA.MOSAPRI1.CI                          0     0  3     0  3   3,1
I5.MO.DA.MOCELLI1.CI                          0     0  4     0  4   4,1
I5.MO.DA.MOCELLX1.CI                          0     0  5     0  5   5,1
I5.MO.DA.MOCELLX2.CI                          0     0  6     0  6   6,1
I5.MO.DA.MODIBUX1.CI                          0     0  7     0  7   7,1
***
I5.MO.DA.MODIBUX2.CI                          0     0  8     0  8   8,1
I5.MO.DA.MOENCOI1.CI                          0     0  9     0  9   9,1
I5.MO.DA.MOENCOX1.CI                          0     0 10     0 10   10,1
I5.MO.DA.MOLOTCI1.CI                          0     0 11     0 11   11,1
I5.MO.DA.MOLOTCX1.CI                          0     0 12     0 12   12,1
I5.MO.DA.MOLOTCX2.CI                          0     0 13     0 13   13,1
I5.MO.DA.MOLOTCX3.CI                          0     0 14     0 14   14,1
I5.MO.DA.MOFETS01.CD                          0     1  0     5 14   15,75
I5.MO.DA.MOFETSI1.CD                          0     6  0     6 14   90,15
I5.MO.DA.MOGRFE01.CD                          0     7  0     7 14   105,15
I5.MO.DA.MOHSLD01.CD                          0     8  0     8 14   120,15
I5.MO.DA.MOHSLDI1.CD                          0     9  0     9 14   135,15
I5.MO.DA.MOSAPR01.CD                          0    10  0    10 14   150,15
I5.MO.DA.MOSAPRI1.CD                          0    11  0    11 14   165,15
I5.MO.DA.MOCELL01.CD                          0    12  0    16 14   180,75
I5.MO.DA.MOCELLI1.CD                          0    17  0    17 14   255,15
I5.MO.DA.MOCELLX1.CD                          0    18  0    18 14   270,15
I5.MO.DA.MOCELLX2.CD                          0    19  0    19 14   285,15
I5.MO.DA.MODIBU01.CD                          0    20  0    26 14   300,105
I5.MO.DA.MODIBUX1.CD                          0    27  0    27 14   405,15
I5.MO.DA.MODIBUX2.CD                          0    28  0    28 14   420,15
I5.MO.DA.MOENCO01.CD                          0    29  0    43 14   435,225
I5.MO.DA.MOENCOI1.CD                          0    44  0    45 14   660,30
***
I5.MO.DA.MOENCOX1.CD                          0    46  0    47 14   690,30
I5.MO.DA.MOLOTC01.CD                          0    48  0    52 14   720,75
I5.MO.DA.MOLOTCI1.CD                          0    53  0    53 14   795,15
I5.MO.DA.MOLOTCX1.CD                          0    54  0    55 14   810,30
I5.MO.DA.MOLOTCX2.CD                          0    56  0    57 14   840,30
I5.MO.DA.MOLOTCX3.CD                          0    58  0    59 14   870,30
I5.MO.DA.MOPERS01.CD                          0    60  0    64 14   900,75
I5.MO.DA.MOREJE01.CD                          0    65  0    69 14   975,75
I5.MO.DA.MOREJEI1.CD                          0    70  0    70 14   1050,15
I5.MO.DA.MOREJEI1.CI                          0    71  0    71  0   1065,1
I5.MO.DA.MOREJEX1.CI                          0    71  1    71  1   1066,1
I5.MO.DA.MOREJEX2.CI                          0    71  2    71  2   1067,1
  *** FREE EXTENT ***                         0    71  3    71 14   1068,12
I5.MO.DA.MOREJEX1.CD                          0    72  0    72 14   1080,15
I5.MO.DA.MOREJEX2.CD                          0    73  0    73 14   1095,15
I5.ES.DA.ESINDX06.CD                          0    74  0    82 14   1110,135
  *** FREE EXTENT ***                         0    83  0    83 14   1245,15
I5.EX.DA.EXSRK201.CD                          0    84  0   101 14   1260,270
  *** FREE EXTENT ***                         0   102  0   276 14   1530,2625
I5.ES.DA.ESSPRK06.CD                          0   277  0   847 14   4155,8565
  *** FREE EXTENT ***                         0   848  0   884 14   12720,555
MVS/DITTO                                    4/23/91 (91-113) 12:06  PAGE     2
---- FILE NAME ---- (SORTED BY EXTENT) ----- EXTENT  BEGIN-END     RELTRK,
***
1...5...10...15...20...25...30...35...40.... SEQ CCC-HH    CCC-HH    NUMTRKS

SYS1.VTOCIX.IMS306                            0   885  0   885 14   13275,15
 *** VTOC EXTENT ***                          0   886  0   887 14   13290,30
SYS1.VVDS.VIMS306                             0   888  0   888  9   13320,10
  *** FREE EXTENT ***                         0   888 10   888 14   13330,5
I5.ES.DA.ESSPRK09.CD                          0   889  0  1297 14   13335,6135
I5.ES.DA.ESSPRK14.CD                          0  1298  0  1400 14   19470,1545
I5.ES.DA.ESSPRK15.CD                          0  1401  0  1773 14   21015,5595
I5.ES.DA.ESSPRK16.CD                          0  1774  0  2128 14   26610,5325
I5.ES.DA.ESMRMI07.CD                          0  2129  0  2584 14   31935,6840
THIS VOLUME IS CURRENTLY  87 PER CENT FULL WITH  3212 TRACKS AVAILABLE
***
```

Looking to this report, we can establish the disk is already organized with the datasets grouped physically in an application respecting order.

3.4 Concluding remarks

Continuing growth in the size and complexity of large systems has created a need to simplify and automate storage management functions and to ensure the performance requirement. We already had to cut the diagnosis phase into two parts, and even at a macroscopic level the amount of data collected start being huge.

The personnal tool proposal must be understood as another way to look the performance detection problems. It's obvious, due to the quantity of storage to control, that it won't be possible any more to establish performance control as depicted, for instance, in the proposed example of this chapter.
We need to emphasize automated methods to partly avoid performance problems and simplify the management.

In the next chapter, we will discuss the new concept Storage Management Subsystem (SMS) has brought; further, we will look its influence over the automated management of I/Os and try to answer to the question 'Can I/O management be automated ? '.

4. Storage Management Subsystem

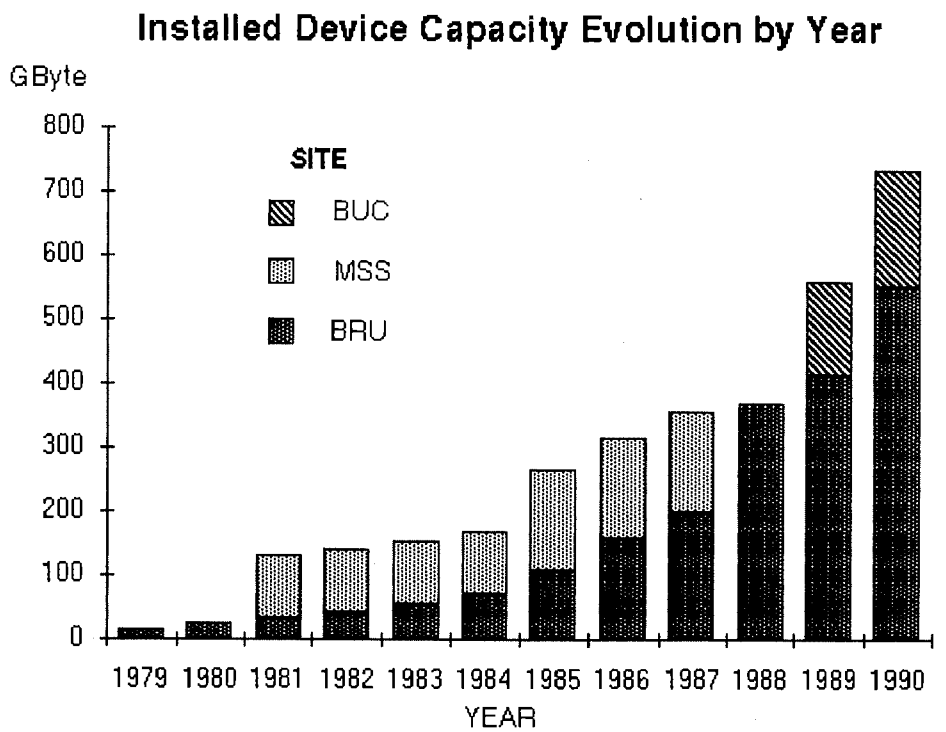
The concept of storage-managed subsystem is an evolutionary one, that separates the logical view of data from physical device characteristics, simplifies interface for the use and administration of storage, integrates the functions of storage management products and provides a synergy of hardware and software functions to effect complex-wide management of external storage resources. The goal of this chapter is to explain the need and improvement require in storage management and the Storage-Management Subsystem solution. We will also describe the storage management constructs.

4.1 Need and improvement require in storage management

Regarding storage management improvement, we can specify some problem areas. There is a need for *timely support* of new devices, tools to simplify the movement of data to those devices, and the ability to fully exploit the new devices and control units capabilities. The ability to explore and exploit new storage technologies with minimal user impact may be included into a data storage management flexibility.
A maximal I/O performance and an optimization of the storage resources use is a continuous need concerning the data delivery to and from applications.

Projecting observed growth gives a 45-6O percent compound growth rate for storage capacity during the last few years. [GEL89]

Figure 4.1


Installed Device Capacity Evolution by Year
GByte
800
700
600
500
400
300
200
100
0
SITE
BUC
MSS
BRU
1979 1980 1981 1982 1983 1984 1985 1986 1987 1988 1989 1990
YEAR

The following figure shows the storage capacity evolution at the CGER (Figure 4.1).The application need and the requirements to maintain historical information for legal or business reasons are mainly the causes of this growth rate. It is clear from this growth that improved system control for the more efficient use of storage and the automation of storage management is necessary.

Finally the biggest need is to simplify the process of *managing* data and storage for end-users and system administrators, and simplify the performance management while the storage growth rate increases.

4.1.1 Consequences of user-managed storage

In a user-managed storage environment, users of the system manage their own data for their application (the logical domain) and the storage devices (the physical domain) on which that data reside. User specification interweave characteristics from both domains, as do device dependencies within applications. This obstructs the separation of the domains for the users, and ensure maximum impact when the configuration changes or data movement must be done for performance tuning.

In this mode of managing data and storage, users may be characterized as having too little information and too much control. The user specifies data storage in explicit device terms. The user is considerably affected when devices are reconfigured or replaced by new device types. To prevent a lack of sufficient storage space and abnormal job endings, the datasets are over-allocated. A consequence of this over-allocation is the wasting space and increasing fragmentation problems when datasets are deleted. It is also the user's responsibility to match the data requirements to storage capabilities. Thus, users must be sensitive to the kinds of devices available (and to their characteristics) to meet the needs of their applications. Users perform also those tasks with the single focus of their own application. A user does not or cannot take into consideration all the other applications that may be run currently. The user has not a global view of the storage device requirements.

The less sophisticated the user is, the more likely the user will select unnecessary space parameters and device types of data. The consequence is that more knowledgeable users will be prevented from getting what is really their needs.

Just following is an over-allocation example :

```
// HARD      DD DSN = K.X06601.D022A1.DATA,
//               DISP = (NEW,CATLG,CATLG),
//               UNIT = SYSTSO
//               SPACE = (CYL,(40,20)),
//            DCB =(LRECL=132,BLKSIZE=23364,RECFM=FB)
```

Without the global information of a storage administrator, the user does exercise over some precious storage resources potentially disastrous consequences on the user's application as a whole.

On the other hand, a storage administrator is somewhat in the opposite situation. He has huge amounts of information available as to the storage and data status, but little control over the on-going process of applications. In many installations, the administrator can do little more than enforce procedures for users of storage, and can be helped by some asynchronous verifications. The administrator is also responsible of the system performance behaviour.

In addition to the user and administrator role, lets consider the role of the system. The system is constrained by the user's explicit specification and by physical device architecture. "Thus, the system is relegated to *only following* orders, with little freedom to decide for itself or override what it has been told." [GEL89]

The role of users, administrators, and the system in a user-managed storage environment creates complexity for the management of data and storage.

Some factors are especially contributing to that complexity. The continuing growth in the area of data processing created for on-line use affects the storage management. The general increased data are not only a consequence of business growth but are also the result of various effect of requiring more data for reporting, database, and for the growing amounts of inactive data for archive, vital records retention, and backup. A fast system disaster/recovery plan is also causing a growth of storage capacity.

In addition to the amount of created dataset, there is an increasing need for better data access in terms of performance.

Both the *data growth* and its *accessibility* place continuous pressure on data and storage management. Performance tuning becomes more and more complex in such a growing environment, and the windows for doing data and storage maintenance become smaller as the need for 24 hours 7 days a week operation start occurring.

4.1.2 Problems and management requirements

The problems introduced by the explained consequences of user-managed storage can be splitted into five areas of data and storage management: space management, performance management, availability management, device install management, and system-wide management.

We will now discuss about each of these management areas.

.1. *Space management* refers to the allocation, manipulation, and control of physical external storage space.
Problems associated with physical space management include out-of-space failures, fragmentation, poor capacity utilization, single-volume constraints, and device-type constraints. Applications and software should be isolated from dependencies on physical device characteristics to solve these kind of problems. Control of storage resource should be centralized, and the system should be empowered to manage that storage resource.

.2. *Performance management* refers to the placement of data on physical storage such that the logical access requirements of the data fit with the access capabilities of the storage devices. Datasets have varying requirements for access performance. In addition, many datasets access depends on the way the application is referencing them, and the planning process of the belonging application. The decision of which datasets have to be placed on which devices is often a difficult process.
The real placement is frequently performed after a problem has occurred, which often creates new performance and/or contention problems.

.3. *Availability management* refers to the process of maintaining the datasets disposition available in an acceptable delivery period, also and especially in a multisystem environment. Managing data availability has been most of the time a manual process, and is also most of the time associated with the responsibility of individual applications or of the installation as a whole. Typically, the preparation for an invalid data or device failures has been accomplished during a backup window, which is usually a low data activity rate period when storage volumes and individual datasets are dumped. The preparation for an invalid data or device failure may also be done using a dual copy process[1] .
There are also growing requirements for maintaining copies of data that are not, or nearly not, expected to be needed again. This includes archival and migrated data, vital records, and data to be copied for disaster recovery and legal purpose.

.4. *Device install management* refers to controlling the impact of attaching new or additional storage devices and making them available for use. The addition of storage to a system has often been disruptive and required changes to applications with *devices dependencies.* This problem of applications with devices dependencies is less disruptive since it has been turned into a pool of devices dependencies.
Anyway, the cost to an installation of converting existing data and storage to new devices is high.

.5. *System-wide management* refers to unified control of data and storage across systems in an installation.

We note that these individual management requirements are not independent. Managing storage space affects the performance of the devices. Managing availability affects the amount of useful space.

[1] Deeper information concerning the dual copy process are available in ANNEXE 1A

"Any solution that attempts to meet an individual management requirement must take into consideration all the management requirements." [LAI90]

The result of this all set of problems is an integration of the various solution elements into a single system manager.

4.1.3 Revolution within evolution

To solve the management problems described above implies the need for a revolution in the way data are stored and managed. The necessary changes would fundamentally affect storage devices, I/O processing, data allocation, data and storage administration, and the mechanisms for relating data requirements to storage capabilities.

4.2 The storage-management subsystem solution

"Storage-management subsystem is the 'synergistic' relationship among software facilities, hardware facilities, administrators, and users that enables the operating system to direct the effective use of external storage." [GEL89]

The separation of logicals and physicals views of storage is one of the basic principles of storage-management subsystem.

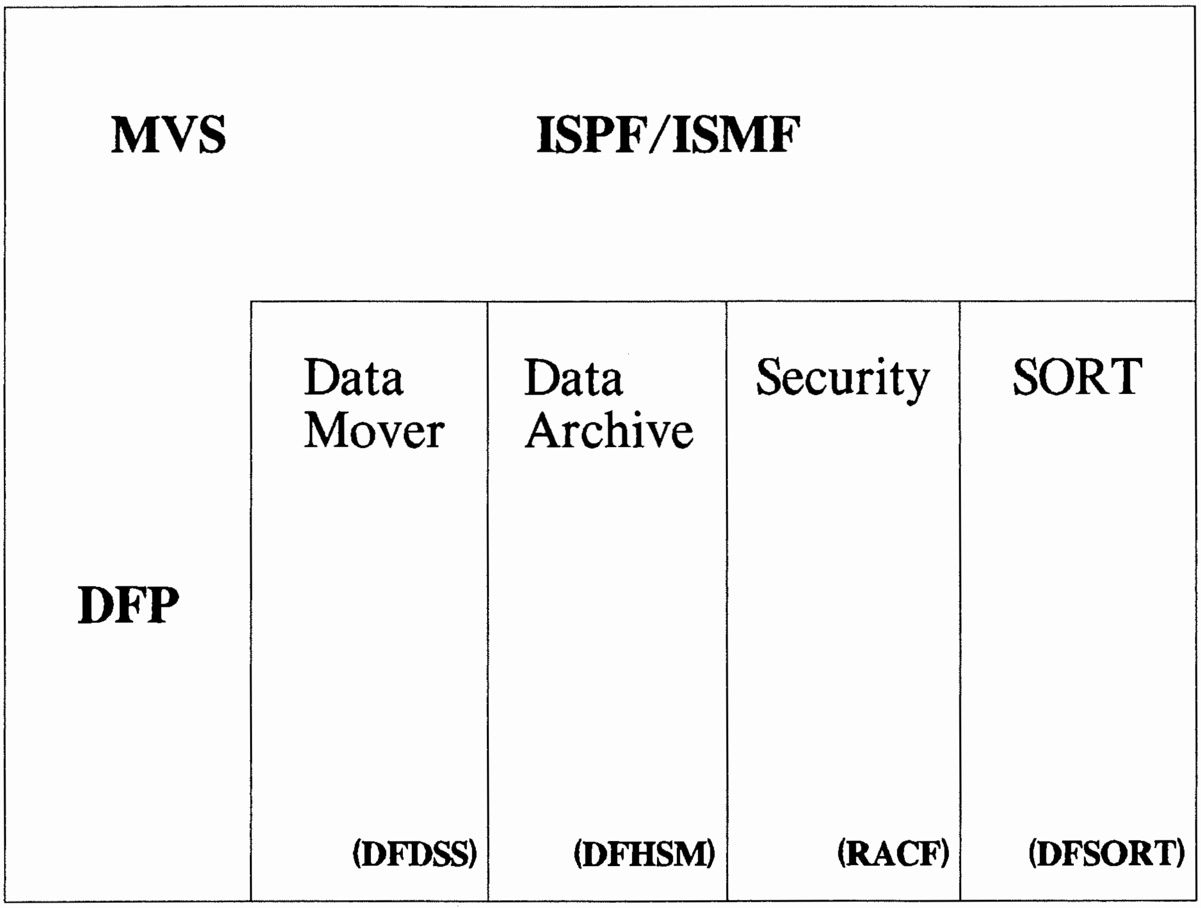
A) THE LOGICAL DOMAIN

The logical domain pertains to the management of data, and includes requirements such as data specification which are record format and record length, data performance which concerns the desired speed of the access, data backup which are the frequency and the versions of the backup, data retention needs which is the retention period, and data security needs which define the type of access and the associated persons concern with it.

B) THE PHYSICAL DOMAIN

The physical domain is concerned with the storage devices themselves and includes characteristics such as capacity, performance, pathing, system connectivity, tuning, physical location, allocation, and free-space maintenance.

Figure 4.2



MVS
ISPF/ISMF
DFP
Data Mover
(DFDSS)
Data Archive
(DFHSM)
Security
(RACF)
SORT
(DFSORT)

Another of the basic principles of storage-management subsystem is the simplification of the user interfaces. Simplification is the process by which users control external storages and the data that reside in it with a less sophisticated knowledge of the system components, and make thus the process of specifying, controlling, and managing both the logical and physical domains easier.

As a subsystem, SMS is also integrating the functions of some storage management products such as MVS/DFP[1] ,HSM[2], DSS[3], RACF[4] as suggested by the following figure (Figure 4.2). Thus using hardware function support in the host (MVS/DFP) and in the storage subsystem is a principle by which functions are streamlined. This includes improving data availability (DFHSM) such as migration, optimizing data transfers (DFDSS), supporting the dynamic modification of control information (RACF), and supporting a big scale sort facility (DFSORT).

These principles form the basis for storage-management subsystem.

4.2.1 Key concepts

In a user-managed storage environment, physical storage awareness rules data management, as suggested by the next figure.
A physical storage environment modification creates a major impact on data management facilities. It also encourages users and applications to become dependent on the physical environment to satisfy data requirements.

---

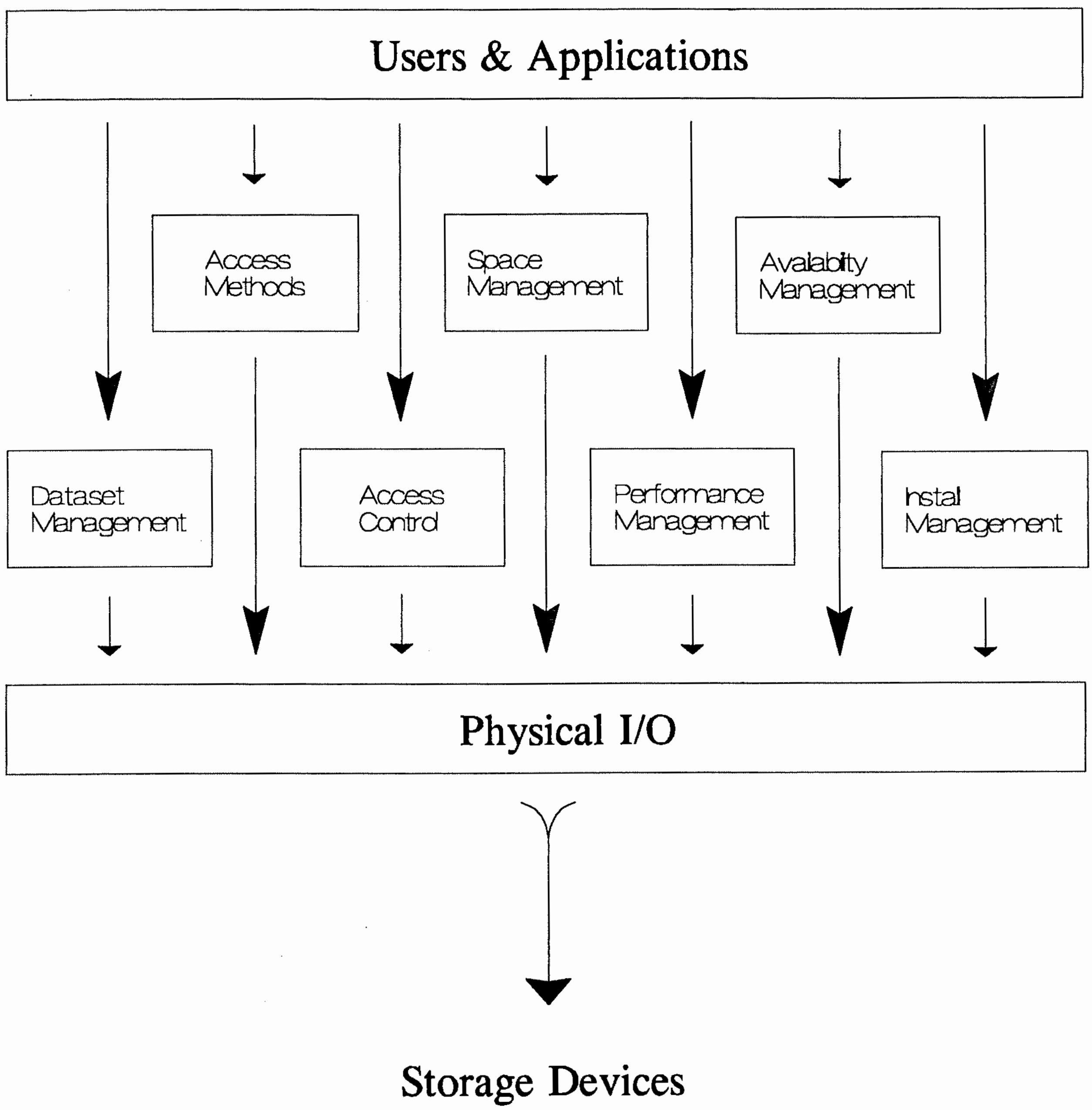
[1] DFP : Data set Facility Product

[2] HSM : Hierarchical Storage Manager

[3] DSS : Data Set Services

[4] RACF : Resource Control Access Facility

Figure 4.3



Users & Applications
Access Methods
Space Management
Avalabity Management
Dataset Management
Access Control
Performance Management
Instal Management
Physical I/O
Storage Devices

The figure shows the relationships among data and storage management facilities, and physical storage in a user-managed storage environment, and the arrows indicate storage awareness (Figure 4.3).

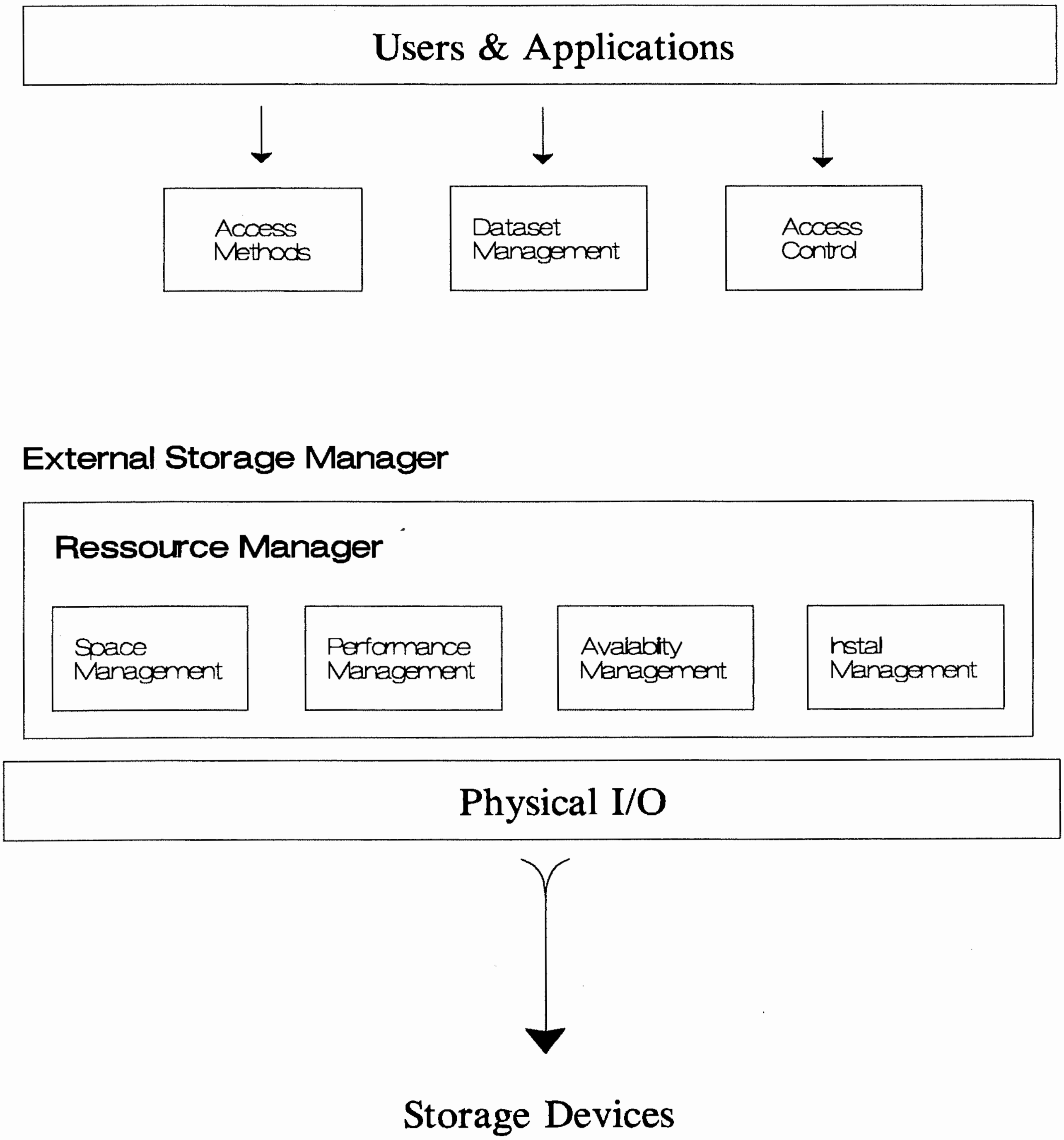
The various facilities include the following functions; *datasets management* such as logical allocation, catalog, and associated services; *access methods* such as record and buffer management; *access control* such as identification and security; *space management* such as physical allocation, utilities, aids, migration and recall; *performance management* such as measurement, reporting, utilities, aids and monitors; *availability management* such as backup, recovery, error detection and repair, and reporting; *device install management* such as storage conversion, data migration on other device, utilities, and aids.

4.2.2 Storage management subsystem facilities [GEL89]

In the design of a *system resource manager* for external storage, two distinct facilities are required. One is an External Storage Manager to manage the physical storage, and the other is a Logical-Data Manager to manage the requirements.

The External Storage Manager (ESM) is the system *resource manager* for physical storage. It is the owner of the physical storage-oriented domain, and is the focus for system management of physical storage with respect to space, performance, availability, and device installation. It also provides support for cache management, control over devices configuration, and multisystem storage control.

Figure 4.4


Users & Applications
Access Methods
Dataset Management
Access Control
External Storage Manager
Ressource Manager
Space Management
Performance Management
Availability Management
Install Management
Physical I/O
Storage Devices

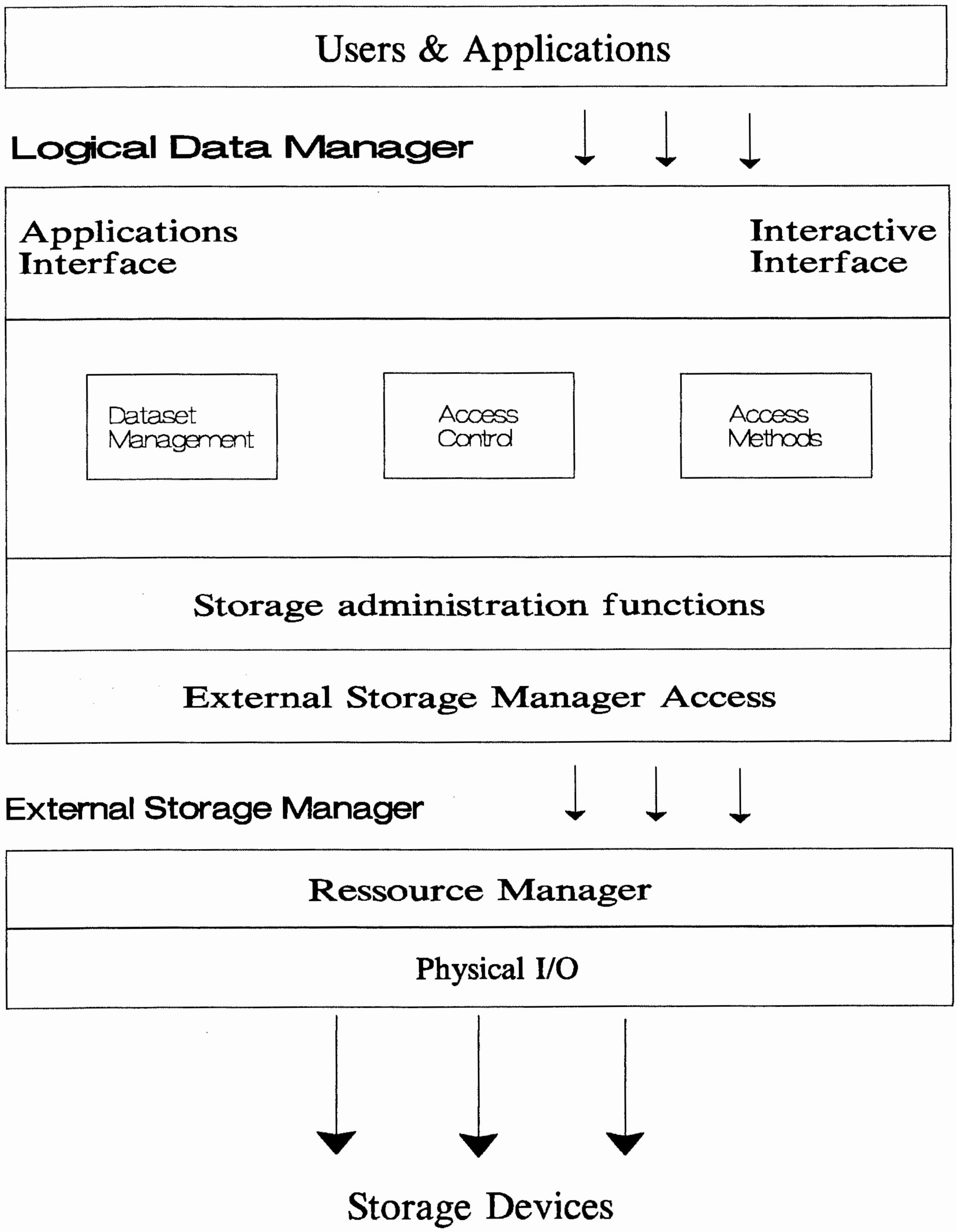
The following figure shows that ESM is the only facility awareness of the physical storage (Figure 4.4). The functions associated with space management, performance management, availability management, and device install management become system managed and *controllable by definitions and policies* established by a storage administrator. Those functions become totally invisible for the user.

To produce an ESM that can control physical storage, it is necessary to remove volume awareness from the user domain, the facilities and the applications used to manage dataset. The user domain and the associated facilities interfaces constitute the domain of the Logical Data Manager (LDM).

The LDM isolates the access methods and dataset management functions from the physical devices.
The LDM is the focus for logical view of dataset format and content, record management, authorization and security, retention and disposition, manipulation and behaviour reporting.
It also provides facilities for the following *storage administration functions*: the definition and maintenance of logical views of storage, the reporting of logical storage status, and the *access to ESM* functions for physical storage monitoring and control.

User applications should be unaware of the existence and function of the ESM. Which means that applications and utilities that have no device dependencies continue to work as they do today; the *application interface* remains unchanged.

Figure 4.5


Users & Applications
Logical Data Manager
Applications Interface
Interactive Interface
Dataset Management
Access Control
Access Methods
Storage administration functions
External Storage Manager Access
External Storage Manager
Ressource Manager
Physical I/O
Storage Devices

The LDM also provides an *interactive interface*, called the Interactive Storage Management Facility (ISMF), for user and administrative access.
The next figure shows the functional components of the Logical Data Manager and its relationship to users, applications, and External Storage Manager (Figure_4.5).

The Logical Data Manager's focusing on the application-oriented domain and the External Storage Manager's focusing on the physical-oriented domain, constitute the components of the Storage Management Subsystem. The ESM and LDM are communicating with each other using a common virtual view of storage.

4.3 Storage management constructs.

Storage-Management Subsystem introduces new concepts and externals to describe and control storage and data, in view of the realization of the objectives of the separation of logical and physical views of storage, and the simplification of the user interface. The developed constructs seek to minimize user awareness and specification of storage, simplify the specification of data requirements for allocation.
Through the use of the constructs, current data and storage use may be simplified and device dependencies may be totally eliminated from applications.

The constructs provided for storage-management subsystem are data class, storage class, management class, and storage group.
Each type of constructs focus over a certain part of the datasets management. Items are associated with each construct. The items characterize the construct they are associated with by their features.

We will now describe the role and the component of the constructs. Once some construct has been established, the migration from an Non-SMS environment to an SMS environment may begin. After migration, all the migrated datasets will be linked with an item of each construct.

4.3.1 Data Class

The Data Class construct represents a template for the allocation and definition of datasets. It provides a mechanism to simplify the externals necessary for specifying dataset organization, record format and length, and block size as well as other parameters used for dataset definition.

The Data Class will also provides some level of project management and control for the common specification for a set of datasets with similar needs. The use of a Data Class prevents from the necessity of specifying individual parameters on data definition. Changes to data-class definitions are not retroactively applied to existing datasets. The Data Class is used only at the creation time of the concerned dataset.

In the current SMS implementation, the following Data Class parameters are included:
- Dataset type (e.g., keyed, sequential, or partioned)
- Record length
- Space requirements
- Expiration and retention dates

The space requirements can be specified in bytes, thereby eliminating device dependencies, such as tracks and cylinders. Several of the Data Class parameters may be specified by users explicitly, overriding those in any associated Data Class.

4.3.2 Storage Class

The Storage Class construct represents the lowest level of storage visibility available to users. It is a logical storage construct. The Storage Class represents the desired level of storage service, based on data requirements. Every dataset that is to be sytem-managed ***must*** be associated with a Storage Class.

The level of service associated with a Storage Class is essentially an agreement between the system and the user. In a certain way, it reflects the performance required of the data in conjunction with the capabilities of the storage in use for the specified data.

The performance service level within a Storage Class represents the response-time requirement of the datasets associated with the class.

In the current implementation of SMS, performance objectives are defined through parameters indicating the following:

- Desired millisecond response time for data using direct access
- Direct access READ/WRITE biasing, reflecting whether the data are more frequently read (as with catalog) or written (as with data base logs)
- Desired millisecond response time for data accessed sequentially
- Sequential access READ/WRITE biasing

Availability forces a certain degree of reliability and accessibility, and reflects the degree to which the system can provide access to data in the light of the physical chosen support. In the current implementation of DFSMS, availability is specified as either STANDARD or CONTINOUS. The first one indicates that normal procedures are adequate and the second one indicates that duplexing via hardware copy, are required.

Through defaulting mechanisms to be discussed later, users are not required to have even an awareness of Storage Classes. However, they may be able to know which storage class is selected for a specific dataset.

4.3.3 Management Class

The Management Class construct represents the criteria by which the life cycle of datasets is managed. Specially, the Management Class contains the policies that control data migration, backup, and retention. The migration and retention policies obviously also contribute to the management of space.

The life of a dataset may be splitted into one of the following states: the *active state* means that the dataset is available to an application for processing; the *less active state* means that the datasets might be migrated and the *inactive state* means that a backup of the dataset has been processed or an archival has been made.

. Datasets that are *active* reside in designed volumes. These volumes are often called primary volumes. The active datasets are directly accessible to applications, and are referenced through Integrated Catalog Facility (ICF).

. *Less active* datasets that are migrated, are moved to secondary volumes (tape managed by robot for instance), though they are still considered active. While migrated datasets are still referenced through the catalog (ICF), the system must stage them up to primary volumes for application access.

. *Inactive* datasets are considered as outside of storage-managed subsystem. Manual intervention (e.g., user command, volume mounting) is required to restore inactive data and reallocate the dataset to primary volume. They are not referenced via the catalog of active data.

The definition and maintenance of Management Class is supported by the storage administrator. Changes to Management Class definitions retroactively affect active datasets associated with that class. In the current implementation of SMS, management class parameters describe the following:

- Migration stage options associated with primary and secondary volumes concept
- Expiration and retention date defaults
- Backup frequency, number of versions, and copy retention
- Partial space release control

As with the two previous constructs, it is not obvious that an installation will define a large number of management classes. The majority of datasets should follow relatively few standard management controls throughout their life.

4.3.4 Storage Group

A Storage Group represents a dynamic pool of volumes. A volume may belong to one and only one storage group. However, multiple volumes may belong to the same group. Logically, a dataset is constrained to reside in one storage group. The volume associated with a storage group will be managed by the storage management subsystem only.

The content of the Storage Group is determined by the installation. A task of the storage administrator is the definition of these groups. This task is quiet important and called the mapping of volumes to Storage Groups.

There are potentially many reasons for configuring storage groups one way as opposed to another. In general, the mapping should be based on environmental considerations, such as the following: device, string, and/or control unit isolation; physical location and/or security; system accessibility, paths; common power bus; or based on business need. Beside these physical constraints, storage groups may also be created for the purpose of maintaining data isolation. Thus a storage group configuration may be established to keep one kind of data from coresiding with another as disaster protection for instance.

The relationship between a logical service level represented by Storage Class and physical device capabilities contained within Storage Group is a many-to-many mapping (N-N connectivity), established via a dialogue between the storage administrator and the system. The requirements of a single storage class may be met by several storage groups. The capabilities of a single storage group may meet the requirements of several storage classes. Through the establishment of the storage class < > storage group relationship, the system has the power to optimize and tune the I/O in placing data to achieve a best fit from a system throughput perspective. This 'automatic placement' will further be depicted.

In the current implementation of SMS, the storage group contains parameters for describing the following area:

- The set of unique volumes associated with the group
- The status of each volume with respect to the group, which means availability for new allocations and for read/write only, or not available at all
- The set of systems that can access the group
- Space allocation thresholds of the group
- Migration, backup, and dump characteristics

The storage group construct is not specificable or manipulable by programmers or end-users. Indeed, its existence does not directly affect the way users manage datasets, because it is available only to the storage administrator. Changes to storage group definitions may require the movement of data to keep coherent in front of the new parameters and the physical data location.

We may not say that an installation will require few storage groups to be defined. The actual number will depend on the *degree and type* of volumes pooling separation desired or perceived as necessary for multiple reasons.

4.3.5 Storage management general remarks

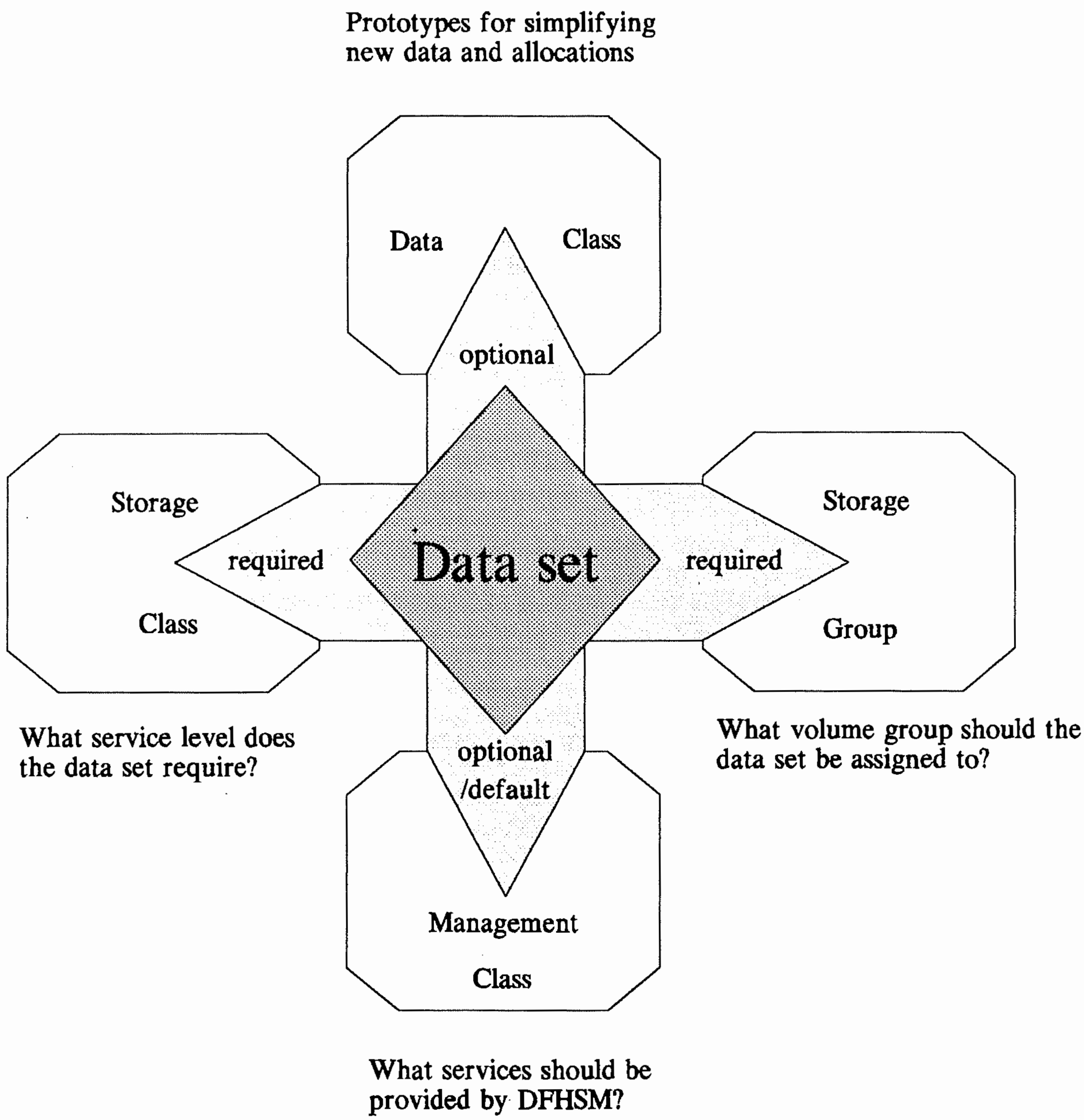
The construct may be summarize as to intend to synthesize the dataset specification. The goal of the constructs definition is to provide a simplifying, unifying, and centralizing approach to the management and control of data and storage. The constructs help to raise the level of data and storage specification of the physical world to the requirements of the logical environment.

Each time a construct definition is created, it has a unique name. Three of the constructs which are Data Class, Management Class, and Storage Class, are externalized to users as parameters, via the Job Control Language (JCL) for instance.

Former dataset allocation

```
//PROM      DD DSN=Q2D.PR.CA.SMS,
//             DISP=(NEW,CATLG),
//             LRECL=80,
//             RECFM=FB,
//             AVGREC=U,
//             SPACE=(80,(500,2500))
//             UNIT=SYSA
```

Figure 4.6


Prototypes for simplifying new data and allocations
Data Class
optional
Storage Class
required
Data set
required
Storage Group
What service level does the data set require?
optional /default
What volume group should the data set be assigned to?
Management Class
What services should be provided by DFHSM?

New dataset allocation

```
//EASY       DD DSN=Q2D.PR.CA.SMS,
//                 DISP=(NEW,CATLG),
//                 DATACLAS=SEQ80
//                 MGMTCLAS=MCBATCHA
//                 STORCLAS=SCSDT
```

Additional control and simplification over construct usage may be achieved by providing increased capabilities for the implicit determination of data allocation for instance. In particular, it will be possible for an inexperienced user to allocate and use data on storage by merely specifying the dataset name. The applicable constructs can be determined by the system based on the specified dataset name.

Throughout the evolution of storage-management subsystem, the constructs will remain durable in their mode of definition and use. It is expected that the attributes, policies, and characteristics within the constructs will change over time as storage-management subsystem evolves but will remain invisible by the users perception.

The following figure summarizes the constructs and their characteristics (Figure_4.6).

4.4 Storage management facilities

Once the storage management constructs have been set up, in support of system managed storage, additional facilities are provided to Data Facilities Storage Management Subsystem (DFSMS). These facilities enable the user, the storage administrator, and the operating system to manage data and storage effectively.

4.4.1 Interactive Storage Management Facility

The Interactive Storage Management Facility (ISMF) is an important concept within the approach to data and storage management that provides a user-friendly interface with SMS. This interactive interface has a two-side purpose. First, it supports a task-oriented use of the functions and facilities. Second, it enables the orderly evolution of facilities.

Supporting the task-oriented use of the functions and facilities is accomplished through a fullscreen, menu-driven set of dialogues to perform the tasks of data management and storage management.
Enabling the orderly evolution of facilities is performed by the provision of a common, consistent set of external commands over the set of products that provide the concept of storage-management subsystem.
A rough sample of the ISMF interface is available in Annexe 4A.

4.4.2 Integrated Catalog Facility

An Integrated Catalog Facility was developed as a consequence of a need for logical data management and storage-management subsystem to locate datasets independently of the physical devices on which they reside. For that purpose, all active datasets residing on managed storage must be catalogued. The Integrated Catalog Facility (ICF) is the main vehicle through which datasets are catalogued, located, and accessed under SMS.

The implications of forced cataloguing of all datasets prevents the creation of duplicate dataset names.

Catalog determination will be based on dataset names. Due to dataset naming conventions and for forced cataloguing reason of all datasets, the name uniqueness is required. The catalog is the information source needed by other DFSMS facilities on *dataset basis.*
This dataset basis includes the constructs definition associated with the dataset as well as other dataset derived information. This focus on the catalog provides a single source of dataset information.

4.5 Naming convention

The Data Class, Storage Class, and Management Class may be explicitly specified by users for new datasets allocation. However, they may also be implicitly specified based over the dataset name. This means that to simplify the explicit specification of constructs for datasets, SMS offers facilities for implicit construct determination.
As a dataset naming convention exists, a progressive migration path from existing externals to the new constructs is possible without requiring changes into the former allocation order.

Automatic Class Selection Routines

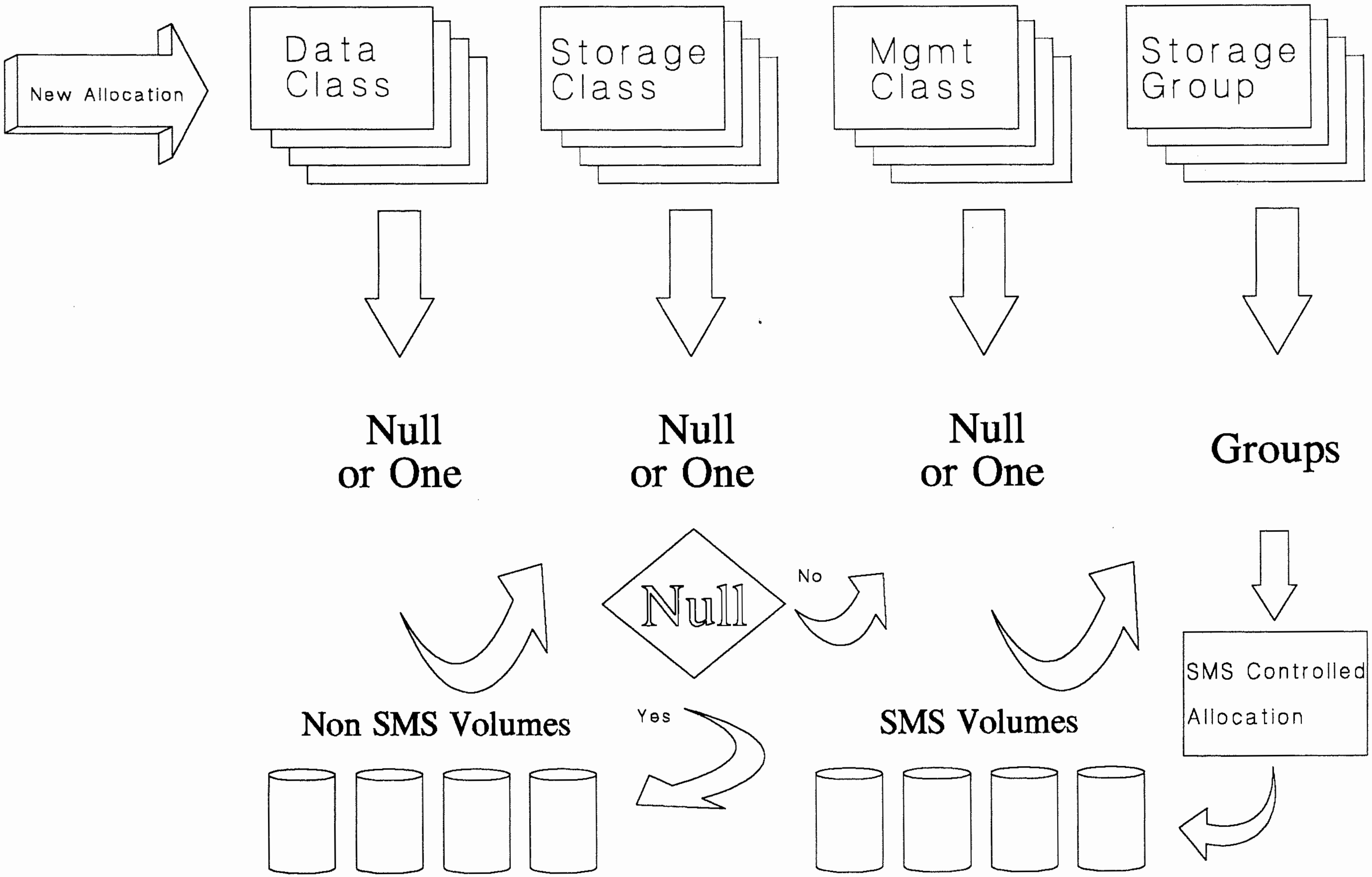


Figure 4.7

4.5.1 Automatic Class Selection

Automatic Class Selection (ACS) is the automatic selection of constructs to be performed by administror-defined filters.

A filter is a collection of specific and/or generic criteria that, when matched, results in the determination of a construct. The criteria's include partially or fully qualified dataset names, job/task and application information, existing Define Data (DD) statement parameters (unit, volume, dataset attributes, etc), and others. The current DFSMS implementation has over thirty variables that can be examined. The specification of the criteria is through a CLIST-like language that provides a construct selection per filter process. There is a filter process for each of the SMS construct type.

This manner, the applicable data class, management class, storage class, and storage groups may be determined.

Here are the execution sequence of Automatic Class Selection routines concerning the constructs determination :

1. Apply the Data Class selection filters
2. Apply the Storage Class selection filters; if no storage class is determined, the dataset will not be system-managed and no further class selection will occur
3. Apply the Management Class selection filters
4. Apply the Storage Group selection filters; at least one storage group must be selected

The next figure shows the sequence for construct determination applying the different Automatic Class Selection routines (Figure 4.7).

Each class selection routine has an associated installation exit that may be invoked for ultimate class determination.

4.5.2 Automatic Class Selection remarks

Automatic Class Selection should be viewed as a conversion tool. As time goes by, the use of old externals should atrophy, and the new constructs used or defaulted take their places. This way, it's possible to migrate datasets through SMS as fast as the analysis for this migration is going on. As a conversion vehicle, Automatic Class Selection represents a toleration or bridging technique to the new externals. We should be able in a long-term to allocate dataset on their name basis. The dataset name may be all that is necessary externally to define a new dataset. This is specially helpful for the majority of users for whom the constructs can be simply determined and who would view their explicit specification as an unnecessary complication.

5. The Storage Hierarchy vs SMS consideration

The management of data within the storage hierarchy was largely a user responsibility. Optimizing the use and exploiting the capabilities of storage devices was completely a reactive task. Improved and additional device technologies have expanded the set of available storage service levels. Storage manager found out that the ability to share and manage the desired service was not highly effective.

With the up-coming of the Storage-Management Subsystem, the logical view of storage enables users to only focus on the *space*, *performance* and *availability* requirements of their data; the system assumes automatically the task of placing and managing the data on storage following as much as possible the user requirements.

5.1 The former hierarchy and its evolution

We already described the conventional way to look at a storage hierarchy based on access speed, cost and capacity criteria, in the first chapter. We will depict some major points which put pressure upon the storage hierarchy evolution.

System administration was kept busy tuning the storage balance workloads and contention and achieved acceptable performance.
Big installations began to become concerned over the changing business application data needs, over the increasing needs of the external storage, and over the people costs to manage both data and storage with respect to space, performance, and availability.

The requirement for 24 hours a day, 7 day a week operation reduced the available windows for performance tuning and backup processing. The use of software tools such as automatic migration facilities (DFHSM[1]) were provided to ease the load of storage management.
The effectiveness of such tools was limited due to user control over data placement on specific devices. Remove user control over storage is a continuous concern to simplify the management.

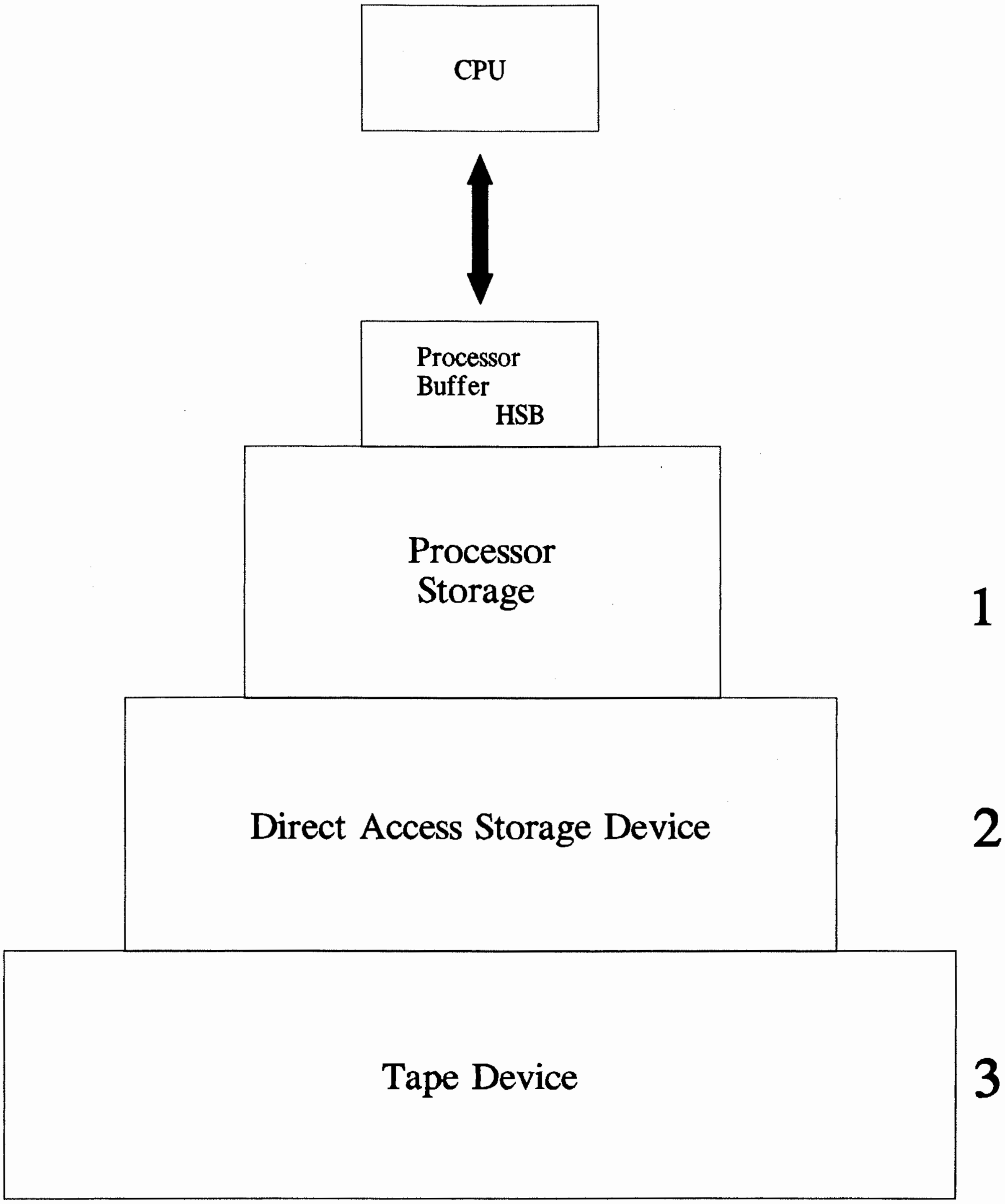
With the introduction of SMS, the software tools providing automatic management have been upgraded. With the introduction of new physical features the so described storage hierarchy demonstrates a continued evolution.

5.2 Dimensions of the storage hierarchy

As described in Chapter 4, the Storage Management Subsystem is a major step in moving from the environment of user management of storage to "system-managed storage". It provides an integrated solution including the already existing software tools to ease the load of storage management, and to separate the logical view of data requirements from the physical view of storage devices. The bound between the concept of logical service levels and storage resources administration is established by a storage administrator through policies. These policies are materialized through the constructs known as data class, storage class, management class, and storage group. *The system uses the logical data requirements within the constructs to select and manage data storage.* [ART91]

[1] DFHSM : Data Facilities Hierarchical Storage Manager

Figure 5.1



Storage hierarchy based on :

. Access speed
. Capacity
. Cost

At the user level, the storage hierarchy can thus be viewed as a 'black box' in which the individual device characteristics and the dataset physical characteristics are known only to the system and the controlling administrator. [GEL90]

The challenge of system management is to optimize the use of the hierarchy with respect of data service requirements.

The users only have a classical view of the storage hierarchy based on speed, cost and capacity and is described as a "top" to "bottom" architecture with three major components which are the Processor Storage, the Direct Access Storage Device, and the Tape device (Figure 5.1).
At the storage administrator level, this view is actually insufficient to characterize the capability of storage *needed* and *provided* to meet data demands. What is "provided" refers to the physical characteristics of the storage. What is "needed" concerns the logical requirements of the data to be placed on that storage.
The storage administrator needs to know what's going on into the user 'blackbox' view.

5.2.1 Physical dimension of the storage hierarchy

The physical dimensions of a storage hierarchy is basically composed of 'what is provided'.

- Tape cartridges that reside outside of direct system control, and are accessed manually
- Tape cartridges library in which mounting the tape may be system-controlled and require no manual intervention
- Direct Access Storage Device which can be system-controlled with respect to data placement and access

- Control Units which manage the bound between the channel and the Direct Access Storage Device
- Processor storage which includes main memory and expanded storage, in other words storage in which data may be placed and accessed without the need of "traditional I/O" operations

After listing some major physical components of the storage hierarchy, it is quiet interesting to look at it with other criteria than speed, cost and capacity.
Some new criteria's were already brought out when the storage administrator started using management facilities such as migration tools (DFHSM).
The 'former hierarchy' may not be considered as a hierarchy any more.

Each type of storage provides particular features following several criteria's.

- Density refers to the amount of data that can be stored in a physical unit of space. Increases in storage density are typically perceived as being good.
- Size refers to the total capacity that a storage type can hold.
- Volatility concerns the ability of a device to maintain data over a certain period of time.
- Removability is a function of control. The notion of managing data that is no longer *immediately* available to the system.
- Performance includes more than simply data transfer rates. In particular, it includes the aspect of "mean time to first byte" that is the time it takes to reach the first bytes of the required data.
- Connectivity criteria is function of the physical configuration. It refers to the accessibility to the device on which the data is stored.
- Reliability focuses on single points of failure, problem determination, and repair time. Dataset under control of a single device (or library) increase the impact of a device (or library) failure.

Given these variety of criteria's, it is quiet difficult to group storage devices into an effective hierarchy concerning the placement of datasets.

If one were to focus on performance, for example, processor memory followed by cache and then DASD, might be the clear choice.
If one were to focus on performance as SMS suggests to do it, we should classify the DASD based over the storage requirements. Under a single DASD we may have various datasets performance service levels, this means we cannot classify any more the DASD into a hierarchy based under a performance criteria. However, as the storage management is made at a dataset level, the datasets themselves may be classified into a hierarchy.
Size constraints may require Tape storage to be at the "top" of the hierarchy, with processor memory at the bottom.

5.2.2 Logical dimension of the storage hierarchy

At the user level, the storage hierarchy can thus be viewed as a "black box"

The logical dimensions of storage hierarchy is function of the available capabilities to meet data requirements. The entire storage supports the logical data attributes of space, performance and availability.

These are the considerations for data placement when viewed from the user level without taking care of the physical dimension.

. Space refers to the initial size and future growth of the data. This logical view is not a function of device capacity or density, but a consequence of application *need* and user specification.
. The performance logical dimension focuses on the speed delivery of data to the application. The performance in this case represents the "acceptable" time lag between the issuing of a request for a set of data and its materialization.

As a logical dimension from the application's perspective, the delivery of datasets to the application is consider without any likely I/O "bottleneck". Data transfer rates, data availability and contention problems are also considered as a toleration of devices capabilities in the front of the respect of the user requirements.

. The availability dimension includes the aspects of accessibility and recoverability. The first one relates to the physical accessibility of storage on which the data resides, while the second one refers to the logical state of the data when accessed and may refer to backup and/or recovery methods for instance.

The logical view of performance ***should*** be stated in terms of data delivery, rather than device capability and explicit data placement.

5.3 Role of storage management

The logical dimension considers the different storage ***design*** that may be constructed at the physical dimension level as a "black box", and ask for a service based on the requirements of data. With this perspective, an application designer can focus on the ***needs*** of the application to materialize and process data, instead of the mechanisms of how the storage subsystem must be manipulated to provide the services required.

The user is isolated from the concerns of the physical domain and can communicate dataset needs purely in the context of logical service levels. As the storage ***design*** evolves technologically, the user's view remains unaffected. The storage administrator, however, can modify parameters within the SMS constructs (or use new parameters as they emerge) to enable the system to exploit new features of the storage ***design*** or reflects changes in the operational or application environments. The storage administrators "are working" inside the so called "black box".

5.4 Concluding remarks

With the described split into the logical and physical domain of the storage *design*, we showed that the logical view of performance must be stated in terms of data delivery rather than device capability and explicit data placement. The link to achieve the splitting is function of the good working of the DFSMS constructs which should respect as much as possible the user requirements.

In the next chapter we will show how SMS try to respect these user requirements by analysing the datasets allocation.

6. Datasets Allocation under SMS

Storage Management Subsystem (SMS) redefines datasets *allocation* and provides, as already explained, a dataset level performance capabilities. These capabilities are exploited by advanced cached control unit functionality. The DFSMS functions supported by algorithms and measurement data sources are poorly defined in the available documentation. They are considered as Object Code Only (OCO) functions by the manufacturer. [ART90]

We will provide in this chapter a detailed description of the allocation process via the Automatic Class Selection (ACS) routines execution.

It would be utopist to say that SMS will find always an exact fit for datasets in respect of the user specification. The dataset's requirements for the logical dimensions which are space, performance and availability against the physical dimension components will be respected as much as possible. It would be hard to properly place each dataset of an installation *manually* and correctly at the first time. The role of the storage management is thus to match the logical dimension requirements of data with the physical dimensions capabilities regarding the system workload.

Since manually performing the link, between the user *needs* and the capabilities *provided* by the storage device, has already been recognized to be unfeasible at a large scale. It is imperative that the system itself manages this process. In the past, the system provided considerable information available (such as RMF[1] and SMF[2] data) but little ability to act on it, whereas the user had total control over data placement.

---

[1] RMF : Resource Management Facility

[2] SMF : Systems Management Facility

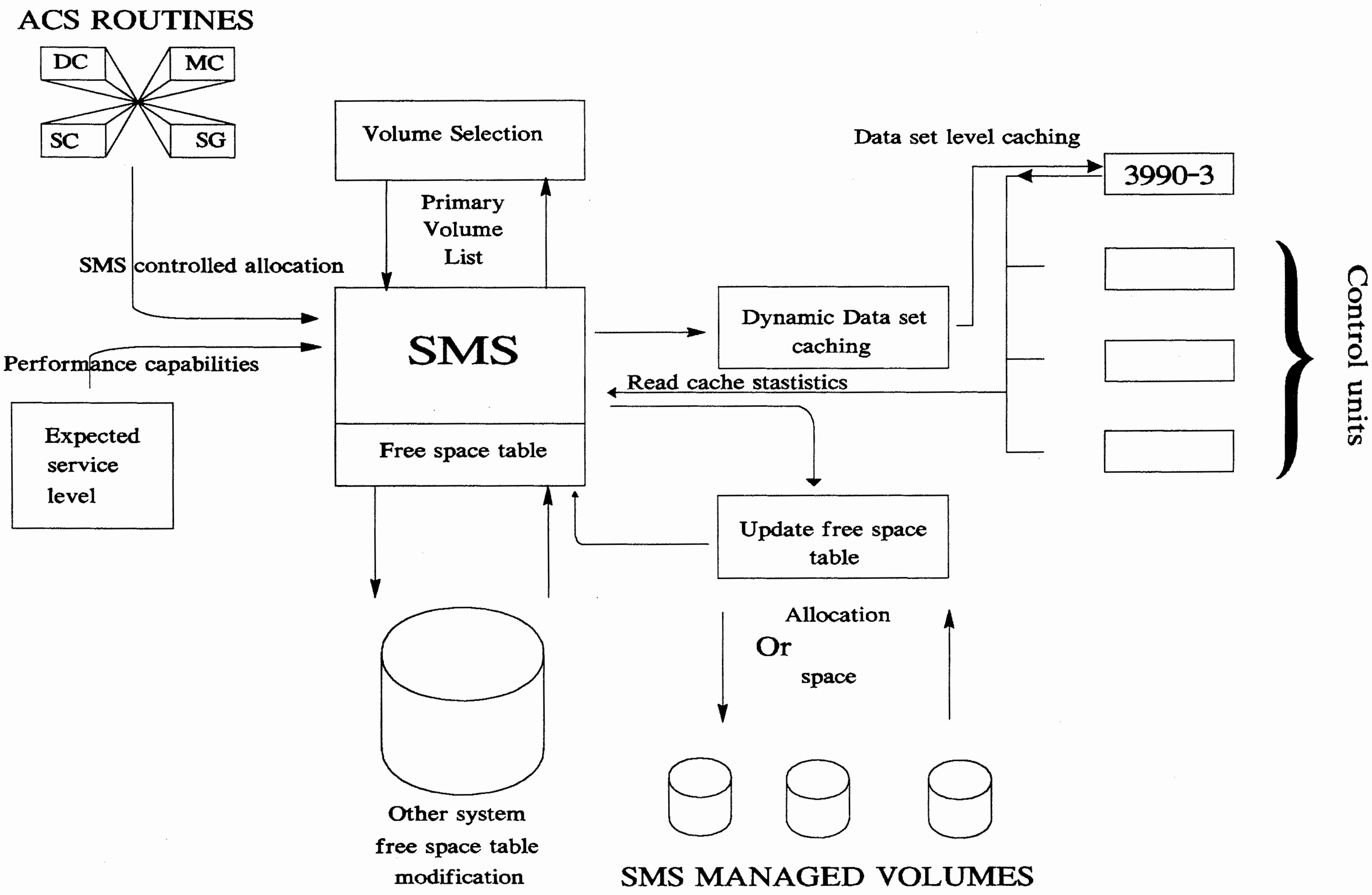

ACS ROUTINES
DC
MC
SC
SG
Volume Selection
Primary Volume List
SMS controlled allocation
Performance capabilities
Expected service level
SMS
Free space table
Data set level caching
3990-3
Dynamic Data set caching
Read cache stastistics
Control units
Update free space table
Allocation
Or
space
Other system free space table modification
SMS MANAGED VOLUMES

Figure 6.1

The system administrator was relegated to react via exceptions behaviour (such as performance problems detection).
With DFSMS, the storage administrator helps by the ACS routines specification, controls the vital link between logical user requirements and physical characteristics during the allocation of any datasets. This vital link is also concerned by the performance environment.
The key constructs of DFSMS that provides this "glue", this vital link are storage class and storage groups.

6.1 allocation process

The following figure provides an overview of how datasets are allocated and how datasets performance are considered under SMS. The objective of this section is to explain the allocation steps detailed in this figure (Figure 6.1). The reader should note that this figure represents partially H. Pat Artis conception of SMS allocation process rather than a formal product figure that has been introduced by IBM.

We will first describe the major components of the figure.
The upper left hand corner of the figure represents the execution of the ACS routines. The ACS routines determine the allocation that has to be system managed, and also determine a list of candidate storage groups function of the dataset service level required. SMS with all these parameters identifies 2 lists of volumes for the allocation process.

- The first list, called the primary contains all the volume names in the candidate storage groups that meet or exceed the service level objective for the dataset and have adequate free space to service the allocation.

- The secondary list (not shown on the figure) contains all the volume names in the candidate storage groups that have adequate free space for the allocation, independently of the service level associated with.

SMS bases its allocation on the expected values of the read/write service times for the volumes, for a variety of device type and control unit combinations, rather than measuring the actual performance of the candidates volumes. These expected service level values are available in an Expected Service Level module as it is shown in the lower left hand corner of the figure(Figure_6.1). Furthermore, an ongoing estimate of the free space on each volume is maintained in the SMS address space. The free space on each volume is calculated every time a new dataset or extent is allocated, and the SMS free space table is updated. However, the space value maintained in the SMS address space is an ongoing estimated since other systems that share the volume may also allocate datasets. To address this issue, multiple systems synchronize their free space tables every 15 seconds by reading and writing their own free space table modification, as it is shown in the bottom centre of the figure(Figure 6.1).

After the primary and secondary lists have been depicted, the primary list is passed to the Volume Selection module as it is shown in the upper centre of the figure. The Volume Selection module selects a volume based on the ***instantaneous*** load measured for each of the suggested Control Unit associated with the volume's primary list.

If all of the primary volumes fail for a lack of spaces, the volume with the greatest free space is selected from the secondary list. When the dataset is allocated, an Update Free Space module updates the table in the SMS address space as it is shown on the bottom right hand side of the figure(Figure_6.1).

6.2 Open process

For datasets allocated behind 399O-3 control units, the Dynamic Dataset Caching module is used to manage each of the cache resource while a dataset is opened. Based on the MiliSecond Response time (MSR) target specified for each dataset in its Storage Class description, the dataset receives a status of never, may, or must-cache. These status are communicated to the Dynamic Dataset Caching module, which function of the received set of status modify the control unit cache management.

Briefly, SMS interrogates each 399O-3 on a periodic basis to determine the cache management characteristics. Based on these samples, the I/O specification for may-cache and never-cache datasets are modified to dynamically maximize the use of the cache and to improve the performance of may-cache datasets while attempting to insure the response time of the must-cache datasets.

6.3 Device selection within the allocation process

The selection of a device for the allocation of an SMS controlled dataset is a complex process. In this section, we will examine how SMS determines the *performance capabilities* of the devices it manages.

6.3.1 Allocation process information

As described in Chapter 4, during an allocation process the ACS routines execution associates for each dataset, that has to remain under the control of SMS, a DATACLAS a MGMTCLAS and a STORCLAS value, and two list of candidate volumes for the physical allocation. Once a volume has been chosen, in addition to the dataset itself special SMS information are also written on the volume. These information are used to control other dataset service provided by SMS or one of its subsystems. For example, the MGMTCLAS is used to control the services provided by DFHSM and the STORCLAS is used by SMS's Dynamic Caching Management.

After initial allocation, the ACS routines are not executed again for the datasets unless the datasets are moved, recalled or restored. This means changing your ACS routines will only influence future allocations, not modify the status or services previously made for datasets that have already been allocated. However, the modification of a specific storage class will influence the future behaviour of some related datasets that have already been allocated.

6.3.2 Expected device performance

SMS bases its allocation on the expected values of the read/write service times for the volumes, for a variety of device type and control unit combinations. These expected values are available in a dedicated module.

The following table provides these values extracted from DFP 3.2[1] version of this module (table 6.1). For example, the last line of the table indicates that the expected direct and sequential read performance for a 339O disk supported by a 399O-3 control unit with the fast write feature enabled, are 1O milliseconds and the expected direct and sequential write times are 6 milliseconds. With the fast write disabled for the device (i.e., never-cache specified), all of the expected service times are 25 milliseconds.

Device Performance Capabilities Table

	Direct Read	Direct Write	Seq Read	Seq Write
3880-13 / 3380	14	25	14	25
3880-23 / 3380	10	25	10	25
3990-3 NFW / 3380	10	25	10	25
3990-3 NFW / 3390	10	25	10	25
3390-3 FW / 3380	10	6	10	6
3390-3 FW / 3390	10	6	10	6

Expected device service times (msec) by device and control unit feature combination DFP V3.2

Table 6.1

The dynamic approach of the initialization is based over a device return-code mechanism. The return-code tells SMS over the capabilities of the device, and its value comes from the described module.

Since some IBM compatible devices, typically respond to a return-code indicating that they are an uncached single density 338O volume, SMS will assign them relatively slow expected values as device performance. SMS will tend to avoid them even if they have very high performance capacities.

[1] DFP : Data Facility Product

As will be discuss in the following section, the most practical way for an installation to exploit such devices under SMS is to regroup them via particular storage group features and carefully control the assignment of the concerned datasets by the ACS routines.

6.4 Concluding remarks

A good knowledge of allocation and open process under SMS is important to understand the hidden performance problems. Hidden because the allocation is automatic and has a major impact on the future datasets performance behaviour.

The storage administrator should be mainly concerned with this automatic allocation on two particular points :
- SMS bases its primary list construction on a theorical service times values for the combination of chosen volumes and their respective control unit, rather than using an historic of the volumes performance.
- The chosen volume, on a primary list basis, is selected among others on the instantaneous load measured of its respective control unit.

To avoid performance problems, the ACS[1] routine conception study is one of the most critical study to achieve. The ACS[1] routine execution is mainly responsible of the 'automatic allocation' by the Storage Class & Storage Group mapping.

[1] ACS : Automatic Class Selection

As depicted under non-SMS environment, when a performance problem occurs it is quiet interesting to look in detail to storage behaviour components. Even if SMS suggests an 'automatic management' of datasets, the storage administrator still needs to know how it works. In the next chapter, we will explain the major changes within the measures interpretation.

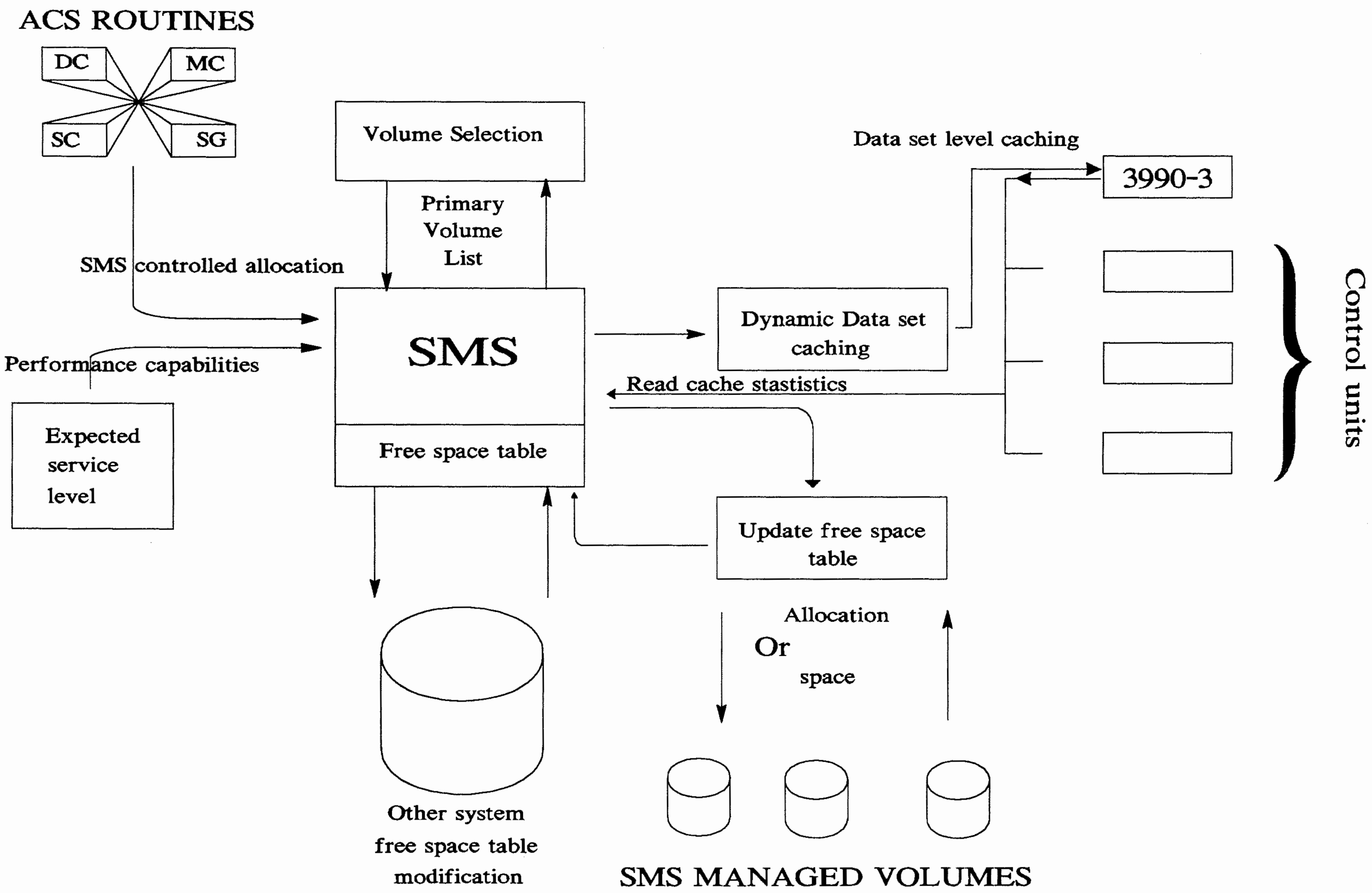
7. Performance problems under SMS

As already described in prior chapters, the storage administrator has specified, in the Storage Class, the MilliSecond Response time (MSR) and some other parameters. The MSR represents the service requirements for the datasets. We should not talk about requirements but merely about wishes. We should talk about wishes because the automated allocation process insure as much as much possible the user requirements.

It's obvious the performance analysts would like to verify how the automated allocation worked, and at least to ensure the good performance behaviour of critical datasets.

We will discuss over the possibilities available to measure the DASD behaviour which are system-managed and the meaning we may give to the collected measures.
Under SMS, all the tools and methods described in the prior chapters are still available and working properly. No upgrade has been made to the DASD measurement tools.

Before getting into interpretation cases, the 399O-3 dynamic dataset caching concept will be explained in details. This dynamic caching method is particularly related with the performance measures interpretation problems under SMS.



ACS ROUTINES
DC
MC
SC
SG
Volume Selection
Primary
Volume
List
Data set level caching
3990-3
SMS controlled allocation
SMS
Free space table
Dynamic Data set
caching
Performance capabilities
Read cache stastistics
Control units
Expected
service
level
Update free space
table
Allocation
Or
space
Other system
free space table
modification
SMS MANAGED VOLUMES

Figure 7.1

7.1 Dynamic dataset caching

One of the overview of SMS dataset allocation and performance management depicted in the following figure is Dynamic Dataset Caching (Figure 7.1). Simply described, SMS manages the 3990-3 cache to insure the performance of must cache datasets and improve the performance of may-cache based on the utilization of the 3990-3's cache and Non-Volatile Storage (NVS). SMS allows the cache's resources to be directed at a dataset level.

7.1.1 Must, may, and never-cache datasets

The storage class returned by the ACS routines specify the target MilliSecond Response time (MSR) for the dataset. At allocation time, this attribute is used to select a candidate volume that meets or exceeds the service level target for the dataset.

When a Storage Class is created, the storage administrator specifies a MSR target between 1 and 999 milliseconds. At DFP 3.2[1] level, the MSR target specification is somewhat less than is implied by the range of available values. Specifically, the numeric range is mapped to three categories.

- Never-cache: a MSR of 999 milliseconds is specified by the storage class,
- May-cache: a MSR of less than 999 milliseconds and greater than or equal to 25 milliseconds is specified by the storage class,

[1] DFP : Dataset Facility Product

- Must-cache: a MSR of less than 25 milliseconds is specified by the dataset's storage class, as depicted in the following table (Table 7.1).

Must Cache Data sets / 3380 Class Devices

Cache Control unit	Read Bias Sequential	Read Bias Direct	None Specified	Write Bias Sequential	Write Bias Direct
3880-13	14 <= MSR < 25	14 <= MSR < 25	14 <= MSR < 25	N / A	N / A
3880-23	10 <= MSR < 25	10 <= MSR < 25	10 <= MSR < 25	N / A	N / A
3990-3 DFW	10 <= MSR < 25	10 <= MSR < 25	10 <= MSR < 25	6 <= MSR < 25	6 <= MSR < 25

Table 7.1

MSR: MilliSecond Response Time
N/A : Non Available

To improve overall performance, all the Integrated Catalog Facility (ICF) present on each SMS managed volume are also treated as must-cache datasets when the volume is supported by a 399O-3.

The table provides the expected MilliSeconds Response time ranges of 3880-23 and 3990-3 cache controllers for variety of read/write and direct/sequential biases. The reader should note that the specification of any write bias characteristics insures that only volumes supported by 3990-3 will be selected for the primary volume list at the time of a allocation. In addition, at DFP 3.2 level, there is no ability to specify the priority of cache utilization by may-cache through the value (i.e., 25 to 998) specified for the MSR. [CHR90]

Despite the current lack of any support of cache priority for may-cache datasets, it should be better to employ a wide range of MSR targets for may-cache to position the SMS implementation for future releases of DFP that may offer cache priority. [ART90]

At the time a dataset is opened, the caching requirement of the dataset is determined and is used by the Dynamic Dataset Caching module creating a token.

The token will be used during subsequent I/O operations to communicate the specific service requirements of the dataset level caching.

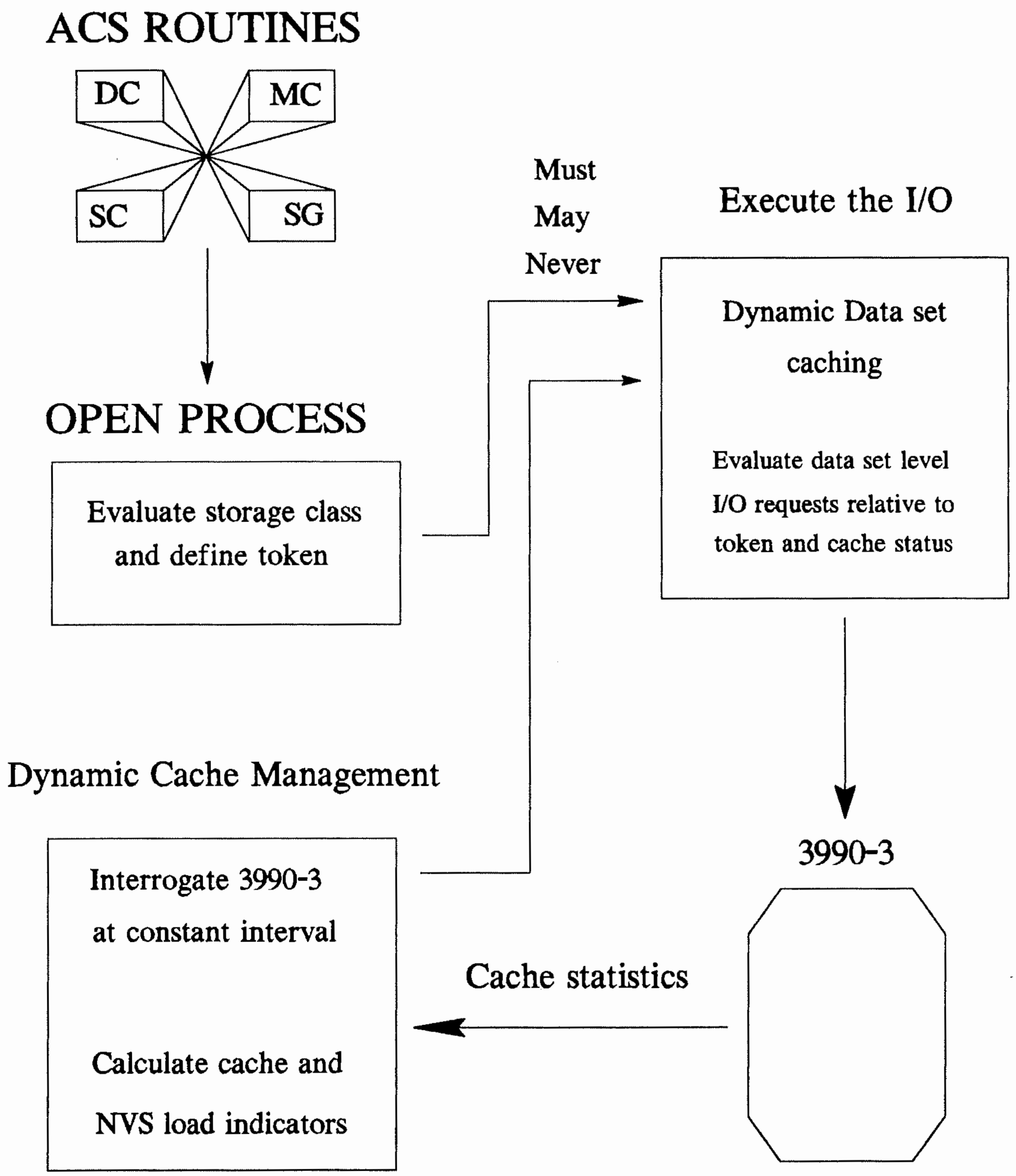
7.1.2 399O-3 Communication

SMS interrogates the 399O-3 control unit every 15O seconds, or at an other specified interval. The process, call Read Cache Statistics, extracts a cache report from the 3990-3 to analyze the performance. The read statistics task calculates :

- The number of system managed devices behind the cache
- The percent may-cache reads allowed to use the cache
- The percent may-cache writes allowed to use NVS
- The percent cache read hits
- The DASD fast write waits per minute.

These data elements are saved in a table for subsequent use by the Dynamic Dataset Caching module which modifies the dataset level caching. Based on the load being experienced by the cache measures, the module can manage the cache's resources to improve the performance of may-cache datasets, and <u>protect</u> the performance of must-cache datasets.

Figure 7.2


ACS ROUTINES
DC
MC
SC
SG
OPEN PROCESS
Evaluate storage class and define token
Must
May
Never
Execute the I/O
Dynamic Data set caching
Evaluate data set level I/O requests relative to token and cache status
Dynamic Cache Management
Interrogate 3990-3 at constant interval
Calculate cache and NVS load indicators
Cache statistics
3990-3

7.2 Dynamic Caching routine

The following figure provides an overview of SMS dynamic caching. As is shown in the figure, a token is created at open time based to communicate dataset service requirements to the Dynamic Caching module (Figure 7.2).

In the event that the read cache statistics task determines that the cache and/or the NVS[1] is over loaded, then I/Os for may-cache datasets cannot use the cache resources.
Relatively few details are available concerning the specific algorithms that implement dynamic cache management.
Access for may-cache reads to the cache resources is based over the read cache statistics task. Access are allowed when the percentage returned by the read statistics task exceeds 70%. Access for may-cache writes is based on the number of DASD fast write waits per minute. As long as this statistic is less than 60 fast write waits per minute, may-cache writes are allowed to use NVS. [ART90] [AMD90]

While the specific characteristics of the dynamic cache managements algorithms are not known, the defaulting sampling interval of 150 seconds is long when compared to the expected duration of a dataset utilization (i.e., dataset open to close time).
Dynamic Dataset Caching will often establish its decisions based on the prior behaviour of the 3990 rather than the specific characteristics of the active datasets access.

---

[1] NVS : Non Volatile Storage

The statistics passed by the 3990-3 describe the performance of the cache at a high level rather than the access characteristics of the individual datasets using the cache. Therefore, dynamic cache management can be awarded there is a problem in the cache, but cannot determine the specific dataset which is causing the problem. Since any actions are possible to solve such problems (i.e. dataset(s) that are overloading the control unit), dynamic cache management will tend to reduce the use of the cache resource by all may-cache datasets.

It is imperative that dataset level analysis tools be used to verify that must-cache datasets have desirable cache characteristics. That is, datasets that have good locality of reference as well as an acceptable write percentage depending on the types of control units that are available. Only one dataset under must-cache feature may cause damage to the general control unit behaviour.
It is recommended that must-cache datasets be identified on a dataset by dataset basis since 399O-3 control unit remains a precious resource in most installations.

The Storage Class definition and the ACS routine are one more time showing up their importance.

As no guarantee of the service requirements are provided, it is the responsibility of the performance analyst to insure the performance of critical datasets. All the explained methods must by consequence be used as prevention studies and problem detection studies, but we need to insure that the collected measures are still interpretable under SMS.

7.3 DASD Measure interpretation

The goal of this section is to point where the meaning given to the collected measures is not respected any more. Unfortunately, this problem occurs using the new cache control unit technology. The following reasoning depicts different configuration cases of volume under control of a 399O-3. [RAN90]

7.3.1 Measure interpretation - Basic case

Considering a set of datasets, all of them are depicted by the same Storage Class. Each time one of these datasets will be opened, they will be considered equivalent by the dynamic cache management.
The collected measures stay independent from the dataset behaviour vs the cache management. The collected measures keep their meaning in spite the datasets are system-managed.

7.3.2 Measure interpretation - Primary case

Considering a set of datasets, all of them are depicted by two different Storage Class. Each time one of these datasets will be opened, they will be considered differently by the dynamic caching management module.
Suppose the MSR in both Storage Class has been specified as 10 and 999 milliseconds. The first one will be interpreted as a must-cache and the second as a never-cache.

While both Storage Class definition characterize the studied volume, the collected measures are not independent any more from the datasets behaviour vs the cache management. The collected measures are function of the use rate of either one storage class definition.

Just taking a look to an obvious example. Let's say a must-cache supported dataset has an average response time of 10 milliseconds, and a never-cache supported dataset has an average response time of 30 milliseconds.

Supposing there is 100 I/Os on the studied quarter

Must Cache I/O	Never Cache I/O	Calculation	Reported measure
80	20	$\frac{80 * 10 + 20 * 30}{100}$	14
20	80	$\frac{20 * 10 + 80 * 30}{100}$	26
100	100		

Table 7.2

The collected measures don't keep their entire meaning, because the given response time is directly function of the occurred I/O type and their associated activity rate.

The problem might not be so deep, if the performance analyst knows more less the volume components behaviour, such as only a few I/O are must-cache and rarely used.

7.3.3 Measure interpretation - Final case

Considering a set of datasets, all of them are depicted by three different Storage Class. Each time one of these datasets will be opened, they will be considered differently by the dynamic caching module.
Suppose the MSR has been specified as 10, 40, and 999 milliseconds. The only difference with the Primary case is the introduction of a may cache feature. While a dataset under a may-cache specification is opened, it will be turned either as a must or never cache through the dynamic caching management. The cache status of this dataset may change during its use.

Now the interpretation of the collected measures is a really deep problem because in none utilization cases, the performance analyst is able to deal with the collected measures.

7.4 Suggested data collector

The major problem concerning the available monitors is, they are not able to manage the differences between I/O types.
Even if they were able to differentiate the I/O at the datasets open process, this won't be enough to guarantee the interpretation in all cases of the measures; because some datasets under may-cache specification may change their I/O type features during their use.
The I/O types identification should be performed at the Dynamic Dataset caching module, because this module dynamically changes the physical order given to the control unit.

The data collector should be able to report the collected measures under two ways at a volume level.

7.4.1 Physical level report

A physical level report means splitting if necessary the current available measures in cache or non cache I/O type.
If a dataset with a may-cache definition is supported by the studied volume, this dataset behaviour will influence the cache or the non cache I/O type, or both measures.

7.4.2 Logical level report

A logical level report means splitting if necessary the current available measures in storage class type. If must-cache defined datasets exist on the studied volume, these datasets behaviour will be the only one to influence their storage class measurement. This also means the may-cache defined datasets won't influence the must-cache defined datasets measurement as depicted at the physical level report.

7.5 Concluding Remarks

The suggested data collector working, at a physical and logical level, on a full time basis is not required for all system-managed volume.

Overhead may occur with such a collector, even if we are not intercepting all the I/O as GTF[1] does. Overhead will be directly related to the data collector implementation.

Nevertheless, even if desadvantages may occur, the release of such a tool is pressing while some measures interpretations are impossible to be performed in certain cases.

This all problematics of interpretable measures, and dynamic caching enforce the importance of SMS pre-study.

In the next Chapter, we will take a look at the most important features to provide a good SMS migration, in regard of the DASD performance.

[1] GTF : Generalized Trace Facility

8.2 Naming convention [ART91] [CGER90]

The major management issues under SMS, is the 'how to achieve a correct SMS migration ?'. The main basis of the migration is the naming convention.
It is also true that the issues associated with establishing control of datasets through effective naming conventions already exist without the specific constructs of SMS.

The purpose of a set of naming constructs is to provide information about the dataset that goes beyond just a unique identifier. This way, information about the datasets, such as ownership, type of data, production vs test, can be gleaned without pursuing information from other sources. Such a simple concept !

There must be an effective naming convention set in place so that dataset allocations can have guide-lines to follow. A good naming convention is absolutely required to process and to ensure the process of an SMS migration. The naming convention is designed to be effective at categorizing datasets for *all* the purposes of storage management. The system should be able to apply filters or use algorithms to determine the storage management actions to be used against the dataset name.

A dataset name should be splitted into an High Level qualifiers (HLQ), a descriptive qualifier, and a Low Level Qualifier (LLQ). The HLQ requires that management applies the necessary resources and support. The descriptive qualifier requires that the naming convention is significantly rich enough to diagnose the application (the owner) using the data. The LLQ requires that management applies the necessary security information and the logical name of the dataset.

As the ACS routines execution are based over the dataset naming convention and some other second hand information. If a good dataset naming convention is not available, an SMS implementation is probably not achievable in the short term.

8.3 Pooling notion

Any effective storage management function requires a mechanism for categorizing datasets for the purpose of managing the stored data. This issue is certainly not new.

The purpose of volume pooling is to allocate the space resource to various groups of users in an orderly and managable manner.

Each user group has an associated set of attributes, such as backup and size requirements. Pooling volumes is effective at managing these issues, such as application isolation and security. Pooling volumes is also effective at bulking datasets on application and size criteria's.

Some of these pooling issue has already been thought and solved before the upcoming of the SMS architecture.

Under SMS, the storage group is the single connection between the logical view and the physical view used to provide satisfaction of space requirements, and performance requirements via the storage class definitions.

The SMS approach is supposed to provide an automated selection process with the help of the ACS routines. The development of an automated selection process is flourishing. The current approach taken in the ACS methodology depends heavily on the classification of devices upon hardware identification, and location behind cache or uncached controllers.

8.3.1 Respect of performance requirements

Volume selection based on performance characteristics of the hardware, does not seem to do justice to the requirements of providing the best I/O performance for the data. There exists an excellent set of knowledge regarding the management of data placement for performance, and certain key issues must be addressed:

- Active datasets on the same volume will impact seek performance
- Dataset hit cache ratios are dynamic relative to other activity under the controller, such as sharing the same cache memory.

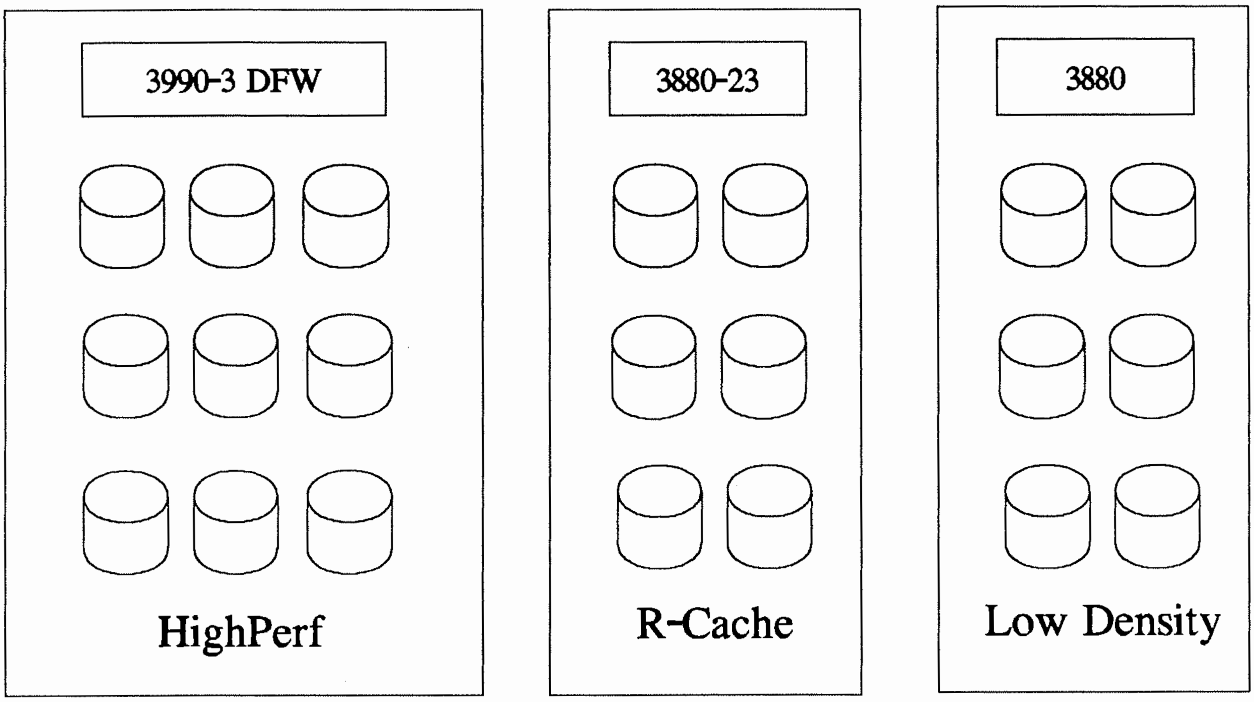
Performance analyst must consider peak time periods over an extensive time horizon in order to evaluate best dataset placement. This means that deep I/O studies for critical dataset has to be done *before* any SMS migration. But this kind of studies are not the achievement of the service requirements. To insure the service requirements, a vertical and horizontal storage pool concept has been brought out.

8.3.2 Vertical and horizontal storage pool

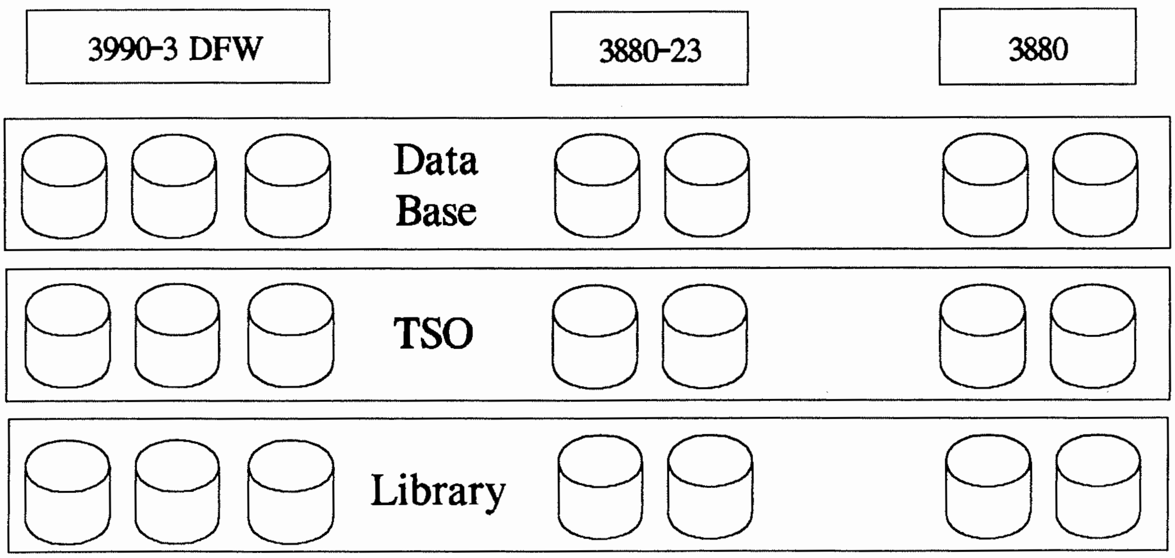
SMS tends to allocate each new dataset on the highest performance alternative device belonging to the primary list. [ART90] We may call this phenomena as a service level escalation.
This factor should be also considered when the storage administrator designs his storage groups, because it doesn't totally ensure the service requirements.

Figure 8.1

Vertical Pools



Horizontal Pools



There are two approaches that have been suggested for the design of storage groups: vertical and horizontal storage pools.

> A vertical storage pools defines a set of like devices behind the same or similar control units.
> An horizontal storage group can include devices with a variety of different features.

The following figure depicts alternative vertical and horizontal pool specifications for 3 strings of DASD (`Figure 8.1`). The horizontal pools are organized by application (i.e., data base, TSO, and library) while the vertical pools are organized by the level of expected performance.
While the horizontal storage pool concept initially appears to be very attractive for allowing SMS to assign datasets in every storage group to high performance devices, SMS will defeat an horizontal pooling strategy by loading all datasets, regardless of the service levels requested, from the top down (in terms of service) within the pool.
The use of horizontal pool definitions will tend to result in the over utilization of high performance devices and the under utilization of older or slower device technologies.

In some cases, the use of vertical pool definitions first, and horizontal pool definitions afterwards should be the best optimized solution for critical datasets under SMS.

If a pre-existing pooling structure before the SMS migration is not available, the SMS implementation will turn to be harder than probably expected.

8.4 Concluding Remarks

A good SMS implementation will provide an easier management than before, but pre-condition elaboration and the implementation studies are not so obvious that we may have thought at the beginning of these explanations.
Tricks have to be used to overpass some SMS lackness and not supported specification within the performance subject.

Four major subject has to be treated *before* starting SMS

- An application oriented naming convention
- An already existing pooling structure
- A deep I/O studies for dataset requiring special resources
- A good N-N mapping, between the Storage Class and the Storage Group definitions

The two first point are considered as general studies, the two last one are performance oriented.

Concluding

The minimisation of the DASD response time which effects on the overall system throughput and end-user is the primary objective of the storage design. The detection of performance problems, the measurement, monitoring and simulation techniques vary considerably as function the objectives of the investigation.

Continuing growth in the size and complexity of large systems has created a need to simplify and automate storage management functions and to ensure the performance requirements.

Storage Management Subsystem is the answer in regard of large systems management concern, by the split between the logical and the physical view of storage. SMS suggests, by its concept, a centralized control of the storage resources via an automated management at a dataset level.

SMS is a quiet complex subsystem that can greatly simplify the role of the performance analyst in the proper placement and maintenance of datasets and inforce the service level requirements.

The use of such a subsystem is not obvious, SMS has the capability to employ a wide range of control unit and device characteristics to meet the service level requirements of datasets. However, insuring the performance of critical datasets remains the responsibility of the performance analyst since the current implementation of SMS provides no facilities to guarantee the performance of individual datasets.

The hardware enables more and more possibilities by its technology improvements. SMS is able to operate with these new hardware features, but a gap is existing between the hardware features and the SMS requirements.

SMS states the logical characteristics of datasets at the allocation level but cannot update them dynamically. Thus it still has to be done manually and *only* by the storage administrator. Some SMS specifications are not taking in account but will be used in further release of some related SMS components.

The dataset allocation process and the non-interpretable measures under particular circumstances required studies for a good SMS implementation.

Lots of work must be done before answering yes to the question 'Can I/O management be automated ?'.

Bibliography

[ART90]
H. PAT ARTIS : from Performance Associates, Inc
Dataset Allocation and Performance Management under System Managed Storage
in : Group for Computer Measurement (GCM) report; 1990

[ART91]
H. PAT ARTIS : from Performance Associates, Inc
System Managed Storage: Planning, Implementation & Exploitation
in : Jaguara Education and consultancy; 1991

[AMD90]
H. DE SMET, M. WOUTERS : from Amdahl
Amdahl information about SMS
in : Amdhal internal report; 1990

[CHR90]
B. CHRISTENSEN, D. DE JONG, PETER Y. HO
MVS/DFP Version 3 Release 2 Dynamic Cache Management
in : IBM International Technical Support Center; 1990

[CGER90]
A. TESSEUR, B. BREMS
Development Control-Standards de Production
in : internal communication at the ALSK-CGER Bank and Insurance; 1990

[COH89]
COHEN, KING BRADY
Storage hierarchy
in : IBM Systems Journal, Vol 28; 1989

[FER83]
D. FERRARI, G.SERAZZI, A. ZEIGNEER
Measurement and tuning of Computer Systems
in : Prentice Hall, Englewod Cliffs (N.J.); 1983

[GEL89]
J.P. GELB
System-managed storage
in : IBM Systems Journal, Vol 28; 1989

[GEL90]
J.P. GELB
The storage hierarchy: Past, present, and future
in : Group for Computer Measurement (GCM) report; 1990

[LAI90]
DAVID LAIRD : from Laird & Stähl Ltd
System Managed Storage - The reality
in : Group for Computer Measurement (GCM) report; 1990

[NEA88]
NANCY NEARING
A method for reporting cashed I/O subsystem performance
in : The Washington Consulting Group; 1988

[RAN90]
S. RANDALL : from LEGENT International
DASD Performance in an SMS environment
in : Group for Computer Measurement (GCM) report; 1990

[SAM90]
STEPHEN L. SAMSON
MVS Performance Management
in : Mac Graw Hill, New York; 1990

FACULTES
UNIVERSITAIRES
N.D. DE LA PAIX

NAMUR

INSTITUT D'INFORMATIQUE

ANNEXE
STORAGE MANAGEMENT SUBSYSTEM ENVIRONMENT IN FRONT OF DEVICE PERFORMANCE

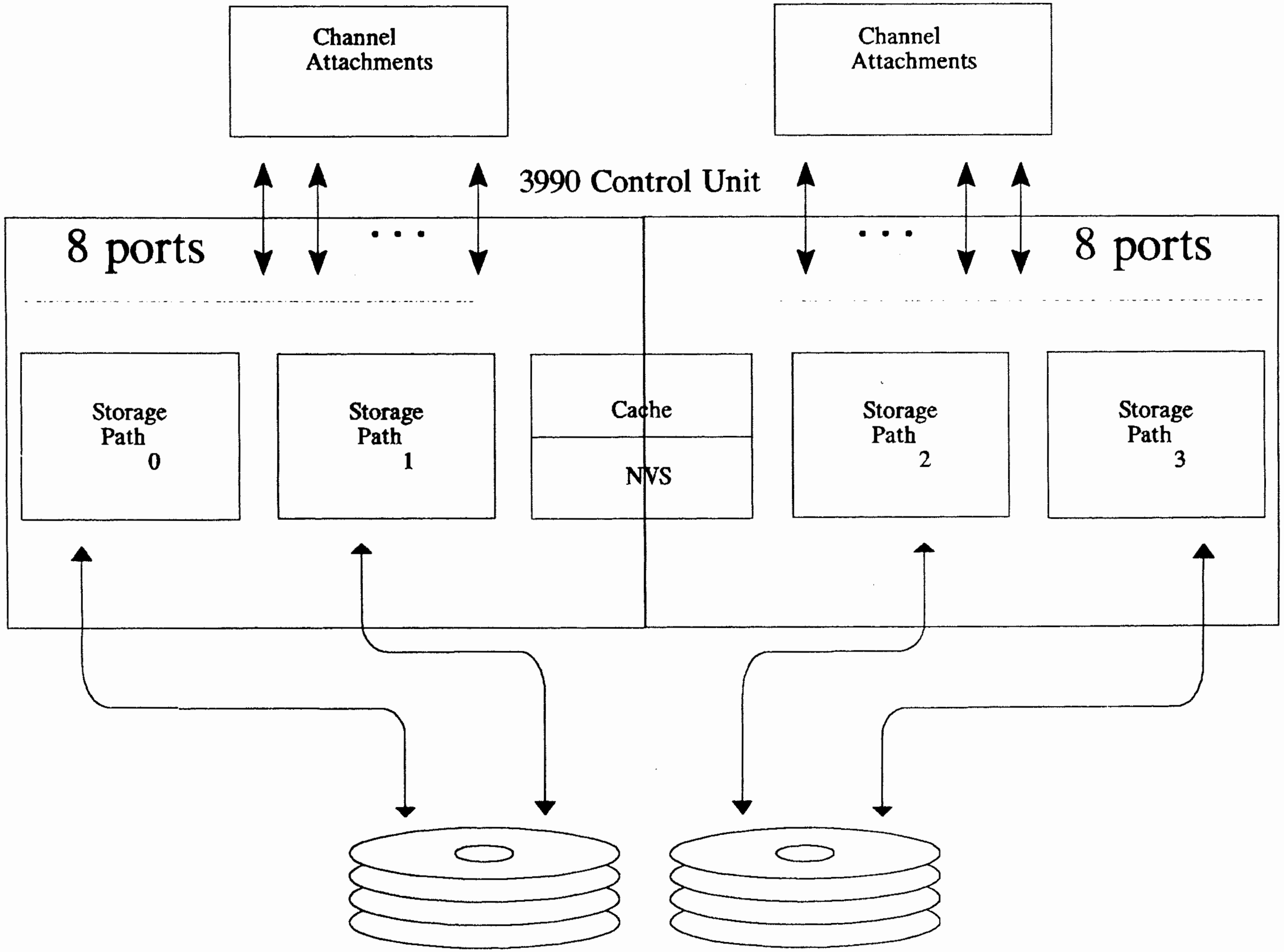
par Marc Parmentier

Promoteur :
Professeur J.Ramaekers

Mémoire présenté en vue
de l'obtention du titre
de Licencié et Maître
en Informatique

Année académique 1990-1991

rue Grandgagnage 21B 5000 Namur (Belgium)



Channel Attachments
Channel Attachments
3990 Control Unit
8 ports
8 ports
Storage Path 0
Storage Path 1
Cache
NVS
Storage Path 2
Storage Path 3

Figure 1A.1

Annexe 1.A

Cache capability within the IBM 399O controller improves performance through DASD fast write and dynamic cache management.

This annexe will try to explain briefly the 399O working.

Each 399O-3 provides two storage directors (SDs), each of which provides two storage paths (SPs). Two SPs form a cluster and share a common power supply. A cluster has attachment capability for eight channels. One control unit can attach 16 channels, eight to each cluster. This configuration is shown in the following Figure (Figure 1A.1).

The Non Volatile Storage (NVS) feature allows write I/Os to be buffered in the controller and I/O interrupt to be processed immediately without confirmation of the physical write. By using the cross-connected NVS features, the system can ensure that failure of one NVS feature is recoverable in the DASD subsystem without requiring any actions by software or operators.

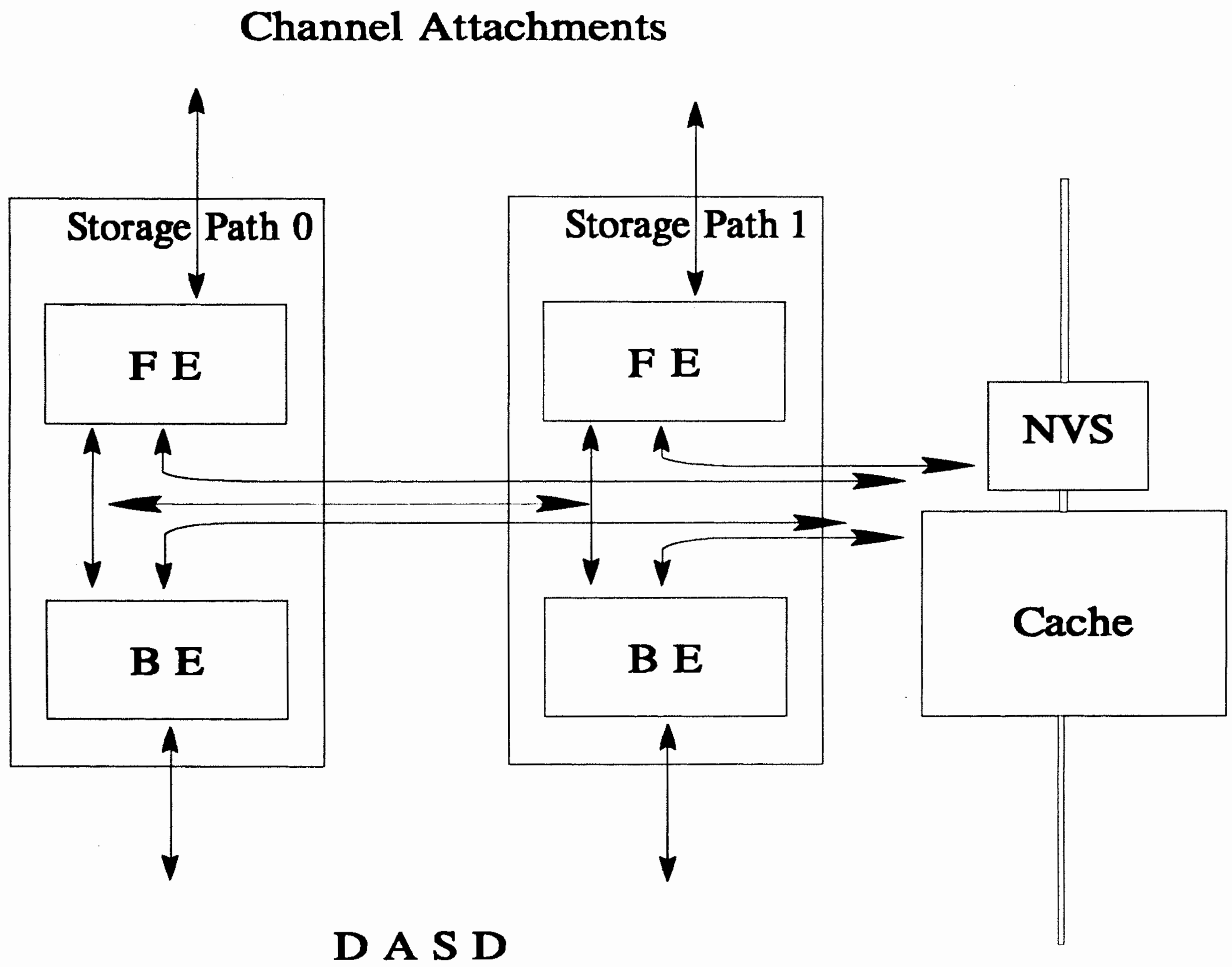

Channel Attachments
Storage Path 0
Storage Path 1
F E
F E
NVS
B E
B E
Cache
D A S D

Figure 1A.2

The dual-copy feature extends that protection to the DASD. Dual copy allows volumes to be backed up on another device from the same string[1] and dynamically maintains an exact copy of the volumes. In the event of a failure, the control unit automatically switches to the backup volume and notifies the system of the failure. The operations staff can then schedule service of the failed device. After repair, the operations staff informs the DASD control unit to reestablish the dual copy. Over a period of time, the good DASD is copied to the repaired device while merging the updates made while the copy process is in progress.

With the 338O conventional control unit, the use of storage for caching is not effective in reducing the number of WRITE I/Os. Read-write ratio is an important parameter in determining the potential impact of large buffer, because only the read I/Os can readily be avoided. The read-write ratio in a large database system is typically of the order of 5 to 1 [ZIM88]. Because most of the I/Os are read operations the buffering can be very effective in reducing the number of I/Os. Even in the case of a one-to-one ratio, buffering can address half of the I/Os and still make a significant reduction in the I/O rate. With the 399O control unit the use of storage is also effective in reducing the number of WRITE I/Os. The following figure shows the Storage Path components with a Front-End and a Back-End processor (Figure 1A.2).

[1] string : couple of DASD behind the same control unit

Annexe 3.A

Monitored Indexes
DASD Reporting : 3380
Resp.time > 30 ms and Act.rate > 2 IO/sec

INQUIRY: DEVRP1 — RUN DATE: 10MAR91
System Id : SYSA — Year : 91

Month	Day	Storage Group	Dev Addr	Volser	Resp Time	Activity Rate	IOSQ Time	Pending Time	Connect Time	Disconnect Time
3	4		C61	BBB100	32.3	9.98	9.2	3.4	1.7	18.0
3	4		74E	HPC300	35.0	4.69	0.1	9.5	16.2	9.3
3	4		874	DBA301	50.2	2.93	29.7	0.3	3.3	16.9
3	5		893	DBA303	32.4	2.89	7.3	0.3	12.0	12.8
3	6		72B	IMS200	41.4	11.16	21.1	0.3	2.7	17.3
3	6		874	DBA301	55.3	3.01	33.7	0.3	3.5	17.8
3	7		323	ATR101	41.2	7.69	19.6	0.3	2.9	18.4
3	7		8EA	ASRT10	33.2	2.58	5.7	0.3	14.9	12.3
3	7		874	DBA301	52.8	3.28	30.5	0.3	4.3	17.8
3	8		874	DBA301	31.9	8.78	12.4	0.3	2.9	16.3

Monitored Indexes
DASD Reporting : CACHED 3380
Resp.time > 15 ms and Act.rate > 2 IO/sec

INQUIRY: DEVRP1 — RUN DATE: 10MAR91
System Id : SYSA — Year : 91

Month	Day	Storage Group	Dev Addr	Volser	Resp Time	Activity Rate	IOSQ Time	Pending Time	Connect Time	Disconnect Time
3	4		84A	IMS302	18.8	2.37	0.8	0.3	2.6	15.2
3	4		842	IMP305	16.2	8.53	2.0	0.3	5.6	8.2
3	4		844	PLA301	15.4	3.56	0.6	6.3	5.5	3.0
3	4		849	IMS301	22.3	2.10	0.5	0.2	2.6	18.8
3	4		85C	IMP307	20.2	11.69	4.6	0.3	7.8	7.4
3	4		85D	PLA300	17.1	5.86	3.4	3.3	8.9	1.4
3	4		85E	IMS308	16.9	2.50	0.7	0.3	2.5	13.4
3	4		85F	IMS300	20.3	13.35	4.9	0.3	2.2	12.9
3	4		851	IMS304	23.8	2.88	3.3	0.4	2.7	17.5
3	4		853	IMS310	17.1	2.01	0.2	0.3	2.3	14.3
3	5		842	IMP305	15.5	7.39	1.7	0.3	5.7	7.9
3	5		85C	IMP307	21.1	11.97	5.3	0.3	8.8	6.6
3	5		85D	PLA300	20.1	6.89	5.7	3.4	9.7	1.3
3	5		85E	IMS308	16.6	2.09	0.6	0.3	2.5	13.3
3	5		85F	IMS300	17.7	10.23	3.3	0.3	2.2	12.0
3	5		851	IMS304	23.4	2.31	3.1	0.4	2.6	17.3
3	6		84A	IMS302	15.5	2.50	0.3	0.3	2.4	12.5
3	6		849	IMS301	21.2	3.32	0.6	0.2	2.9	17.4
3	6		85C	IMP307	20.4	10.12	4.5	0.3	8.4	7.1
3	6		85D	PLA300	16.8	5.91	4.4	2.5	8.5	1.4
3	6		85E	IMS308	16.3	2.18	0.6	0.3	2.8	12.6
3	6		85F	IMS300	18.0	13.40	3.8	0.3	2.1	11.8
3	6		851	IMS304	23.7	2.04	3.3	0.4	3.2	16.9
3	6		853	IMS310	16.7	2.32	0.2	0.3	2.3	13.9
3	7		85C	IMP307	28.3	16.07	13.2	0.4	8.8	6.0
3	7		85D	PLA300	17.8	6.30	5.1	2.7	8.6	1.4
3	7		85E	IMS308	17.4	2.09	0.8	0.3	2.5	13.7
3	7		85F	IMS300	18.7	16.54	5.2	0.3	2.0	11.1
3	7		851	IMS304	26.0	2.06	2.6	0.4	7.0	16.0
3	7		853	IMS310	17.1	2.02	0.2	0.3	2.3	14.2
3	8		842	IMP305	15.7	8.02	2.1	0.3	5.7	7.6
3	8		85C	IMP307	19.1	11.88	4.2	0.3	8.0	6.6
3	8		85D	PLA300	16.9	7.58	5.7	2.5	7.8	1.0
3	8		85E	IMS308	16.6	2.07	0.5	0.3	2.5	13.3
3	8		85F	IMS300	17.6	8.79	2.7	0.3	2.2	12.4

Monitored Indexes
DASD Reporting : 3380
Resp.time > 30 ms and Act.rate > 2 IO/sec

INQUIRY: DEVRP1 RUN DATE: 10MAR91
System Id : SYSB Year : 91

Month	Day	Storage Group	Dev Addr	Volser	Resp Time	Activity Rate	IOSQ Time	Pending Time	Connect Time	Disconnect Time
3	4		DC4	HSM100	36.4	6.09	20.4	0.4	2.8	12.8
3	6		3C7	BBB232	30.0	31.16	8.3	0.4	3.5	17.8
3	6		763	VPW302	37.2	5.93	18.1	0.4	2.3	16.4
3	7		3C3	BBB215	32.8	4.59	6.5	0.4	2.5	23.4
3	7		844	PLA301	40.6	3.88	15.0	0.6	12.9	12.1

Monitored Indexes
DASD Reporting : CACHED 3380
Resp.time > 15 ms and Act.rate > 2 IO/sec

INQUIRY: DEVRP1 RUN DATE: 10MAR91
System Id : SYSB Year : 91

Month	Day	Storage Group	Dev Addr	Volser	Resp Time	Activity Rate	IOSQ Time	Pending Time	Connect Time	Disconnect Time
3	4		844	PLA301 +	19.0	13.61	7.3	1.8	5.9	4.0
3	4		85D	PLA300	17.1	12.84	4.9	2.6	6.4	3.2
3	4		850	BIB300	23.2	2.11	1.4	1.6	7.0	13.2
3	5		844	PLA301 +	15.9	2.15	1.3	1.3	6.1	7.2
3	5		85D	PLA300 +	15.8	12.81	4.9	2.3	6.0	2.7
3	6		844	PLA301 +	27.7	4.64	0.7	0.6	13.7	12.7
3	7		844	PLA301 +	40.6	3.88	15.0	0.6	12.9	12.1
3	7		850	BIB300	21.4	2.58	0.2	0.5	8.8	11.9

Monitored Indexes
DASD Reporting : 3380
Resp.time > 30 ms and Act.rate > 2 IO/sec

INQUIRY: DEVRP1 RUN DATE: 10MAR91
System Id : SYSC Year : 91

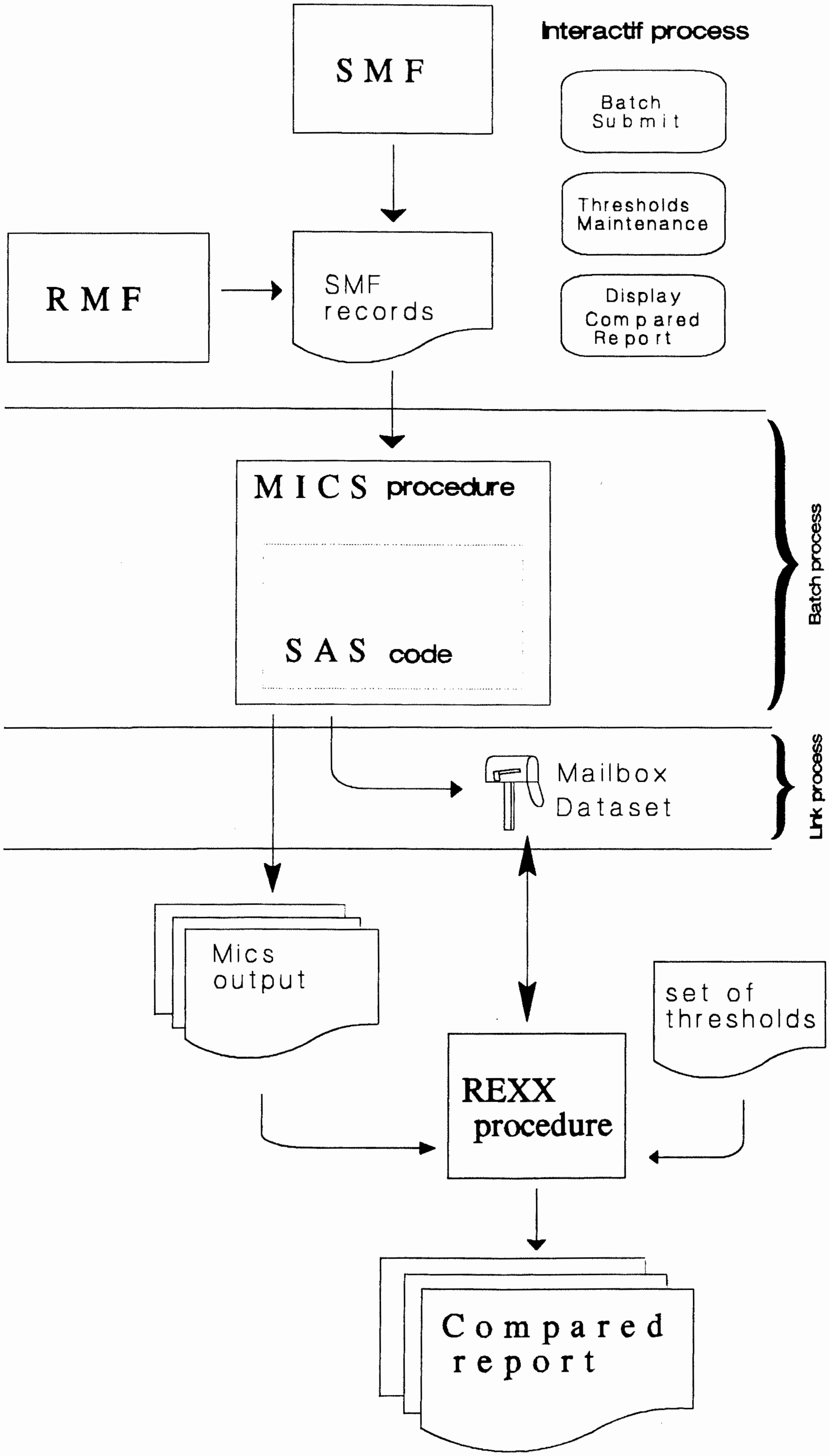
Month	Day	Storage Group	Dev Addr	Volser	Resp Time	Activity Rate	IOSQ Time	Pending Time	Connect Time	Disconnect Time
3	4		1C3	CPAG03	30.1	3.44	0.0	0.5	8.9	20.7
3	4		185	SYSA02	113.7	10.58	95.1	0.4	3.7	14.5
3	4	SGDEVL	870	DEVC02	75.7	17.51	64.3	0.4	4.2	6.7
3	4	SGDEV	871	DEVC00	44.8	27.84	36.4	0.4	2.3	5.8
3	5		1C3	CPAG03	30.4	3.20	0.0	0.7	8.8	20.9
3	5		367	CPAGE2	30.0	3.24	0.2	0.3	8.7	20.8
3	5		87E	IMTC01	34.1	12.26	11.6	0.4	19.3	2.7
3	6		182	SHRA00	36.6	2.26	9.4	0.9	6.0	20.3
3	6		87E	IMTC01	33.2	9.69	10.7	0.4	18.9	3.2
3	7		1C1	HSMA00	36.6	9.60	22.5	0.6	2.7	10.8
3	7		1C3	CPAG03	31.9	2.62	0.0	0.6	10.1	21.1
3	7		226	SPOA01	41.7	16.04	19.3	0.6	4.1	17.7
3	7		367	CPAGE2	31.3	2.63	0.6	0.3	10.1	20.3
3	7		87E	IMTC01	35.1	11.22	10.8	0.4	20.5	3.4
3	8		226	SPOA01	34.4	12.95	13.5	0.6	4.8	15.6
3	8		367	CPAGE2	30.0	2.42	0.6	0.4	8.5	20.5

Monitored Indexes
DASD Reporting : CACHED 3380
Resp.time > 15 ms and Act.rate > 2 IO/sec

INQUIRY: DEVRP1 RUN DATE: 10MAR91
System Id : SYSC Year : 91

Month	Day	Storage Group	Dev Addr	Volser	Resp Time	Activity Rate	IOSQ Time	Pending Time	Connect Time	Disconnect Time
3	4	SGDEVB	CA2	DEVA01	20.9	3.91	6.8	1.2	8.9	4.0
3	4	SGDEVL	CA6	DEVA02	19.3	7.80	7.4	1.3	7.1	3.4
3	4		C4B	BIBA01	18.2	2.68	3.9	1.9	9.6	2.8
3	4		C41	IMTA00	23.6	10.04	7.0	0.7	13.9	2.0
3	4		C44	DBTA00	20.3	6.17	6.3	2.4	6.0	5.5
3	5		C41	IMTA00	20.9	10.30	4.4	0.7	14.2	1.6
3	5		C44	DBTA00	22.3	8.29	7.7	2.3	6.7	5.5
3	6		C41	IMTA00	19.6	9.00	3.3	0.8	13.7	1.8
3	6		C44	DBTA00	21.9	8.22	8.1	2.4	5.7	5.7
3	7		C41	IMTA00	21.1	9.99	4.2	0.7	14.5	1.7
3	7		C44	DBTA00	19.1	8.43	6.7	2.4	4.8	5.2
3	8		C41	IMTA00	16.7	15.48	3.9	0.6	11.0	1.1
3	8		C44	DBTA00	25.0	6.92	8.5	2.2	7.9	6.4

Figure 3B.1



Interactif process
Batch Submit
Thresholds Maintenance
Display Compared Report
SMF
RMF
SMF records
MICS procedure
SAS code
Batch process
Mailbox Dataset
Link process
Mics output
set of thresholds
REXX procedure
Compared report

Annexe 3.B

The goal of this section is to describe the tool process and the panels interface.

The tool process, respecting the three described phases, will be depicted in this section. Concerning the Batch process, an executable procedure will be explained. Concerning the mailbox dataset, some implementation problems will be explained. Concerning the interactive process, a sample of the user-interface capabilities will be shown. The following figure shows the three phases (Figure 3B.1).

A) SAMPLE OF AN EXECUTABLE PROCEDURE

The goal of this sample is to show where the user parameters are used. Each dataset name used, include the day, the system, and the day-period concerned by the process

The procedure is written with the Job Control Language (JCL).
This first step consists to delete the dataset that will contain the MICS report.

```
//*X06601 JOB 93400300N,X06601,NOTIFY=X06601,CLASS=1,
//* MSGLEVEL=(1,1),MSGCLASS=U,USER=X06601
//*TSREF                                                    OUTPUT
WRITER=U240,DEFAULT=YES,JESDS=ALL,GROUPID=D000X066
//DELF EXEC IDCAMS
 DEL K.X06601.D022A1.DATA
 DEL K.X06601.D022A1.DATA NSCR
 SET MAXCC=0
```

The second step consists to call the MICS report environment and to execute directly the included SAS sentences.

```
//MICF      EXEC PGM=SAS,
// PARM='D=MABATCH,DYNALLOC'
//*
//SYSIN     DD  DATA,DLM=àà
 OPTIONS  SOURCE  SOURCE2  NOMACROGEN  NOSYMBOLGEN  NOMLOGIC
NOTES
         OVP NOERRORABEND
         C96
           DQUOTE NOTEXT82
         NOCAPS USER=WORK LABEL NODSNFERR S=72 S2=72
         LS=132 PS=88  ;
 OPTIONS NOSOURCE NOSOURCE2 NOMACROGEN NOSYMBOLGEN NOMLOGIC
NONOTES;
 %LET MICSDBUG=NO; %LET SASVER6=0;
 %LET LS=132; %LET PS=88;
 %MACRO NEGATE; %MEND NEGATE;
 DATA _NULL_;
 CALL MICSLOG('ICF1098');
 RUN;
 %LET  DBID=;  %LET  MOPTS=;  %LET  INQINAME=DEVTST;  TITLE;
FOOTNOTE;
 %INCLUDE USOURCE(#BASMSTR);
 %LET INQGCTLG = WORK.G1;
 %LET _RPTFLE_ = WORK._RPTFLE_;
 %LET _WORK1_  = WORK._WORK1_; %LET _WORK2_  = WORK._WORK2_;
 %LET _WORK3_  = WORK._WORK3_; %LET _WORK4_  = WORK._WORK4_;
 %LET INQUNAME =
          ASLK-CGER MICS SYSTEM;
 %INQTITL(INAME=DEVTST,LS=132,GS=80);
 %LET TMFRHDR = %STR( );
 OPTIONS  SOURCE  SOURCE2  NOMACROGEN  NOSYMBOLGEN  NOMLOGIC
NOTES;
 %LET FILE1 = WORK.FILE1;
 /* MICS File Selection */;
 RUN; %LET FILEREF=DETAIL HARDVA31-01;
 %LET DBID=P; %LET ONEUNIT=Y;
 %LET MWFXDBLS=PDETAIL:P;
 DATA &FILE1;
 SET  %ICFDDNME(ACTION=S,DDNDBLST=&MWFXDBLS,TS=DT,IID=HAR,
      FID=DVA,CYCLES=31-01,ARCHTAPE=N,
      EXECCYOV=N,OLDMACRO=Y,ONEUNIT=N) END=_EOF_;
 %CALLDRV(CCC=RMF,FFF=DVA);
 /*COMMON Data Selection */;
```

```
IF DAY = 22 ;           The DAY depicts the chronological number
                                of the day within the current year
IF ZONE='1'             The ZONE depicts the day-period
   ;
  ;
IF SYSID='SYSA'         The SYSID depicts the studied system
  ;
/* SAS SYSTEM STATEMENTS */;
IF VOLSER EQ: 'AAA101' OR       The VOLSER depicts the device
   VOLSER EQ: 'AMCAT1' OR        name concerned by the needed
   VOLSER EQ: 'AMCAT2' OR               measure
   VOLSER EQ: 'APAGEA' OR
   VOLSER EQ: 'APAGEC' OR
   VOLSER EQ: 'APAGED' OR
   VOLSER EQ: 'ASORT1' OR
   VOLSER EQ: 'ASORT2' OR
   ...      ...      ...
   VOLSER EQ: 'TTT104' OR
   VOLSER EQ: 'TTT105' OR
   VOLSER EQ: 'VPW100' OR
   VOLSER EQ: 'VPW301' OR
   VOLSER EQ: 'VPW302' OR
   VOLSER EQ: 'VPW303' OR
   VOLSER EQ: 'VPW304' OR
   VOLSER EQ: 'XXX100' ;
 /* Data Element Derivation */;
  FORMAT ENDTIME TIME8.;
 LABEL ENDTIME="Time";
 ENDTIME = TIMEPART(ENDTS)
  ;
 %LET FILE2 = WORK.FILE2;
 /* Data Summarization */;
 RUN;
 %LET BY= SYSID  YEAR  MONTH
          DAY  HOUR  ENDTS
          VOLSER
              ;
 %LET BREAK = VOLSER;
 PROC SORT DATA=&FILE1 OUT=&FILE2 NOEQUALS;
 BY &BY; RUN;
 DATA &FILE2; %DVALEN(TS=MONTHS); SET &FILE2;
 %DVASUM;
 RUN;
 /* SAS System Statements */;
 RUN; DATA &FILE2; SET  &FILE2 END=_EOF_;
 DVAAVRES = DVAAVRES * 1000 ;
 DVAAVIOS = DVAAVIOS * 1000 ;
 DVAAVPEN = DVAAVPEN * 1000 ;
 DVAAVCNN = DVAAVCNN * 1000 ;
```

```
 DVAAVDIS = DVAAVDIS * 1000 ;
 DVAAVSER = DVAAVSER * 1000 ;
 /* Structured List */;
 RUN;
 %LET BY= SYSID  VOLSER  ENDTIME
 ;  PROC SORT DATA=&FILE2 OUT=&_RPTFLE_ NOEQUALS;
 BY &BY;
 RUN;
 OPTIONS MISSING=' ' CENTER NODATE NUMBER
           ;
 PROC PRINT DATA=&_RPTFLE_  U  ROUND NOOBS
                              LABEL SPLIT="*";
 VAR SYSID ENDTIME DEVADDR VOLSER DVAACTRT
     DVAAVRES DVAAVIOS DVAAVPEN DVAUTL DVAAVSER
     DVAAVDIS DVASSC DVAAVCNN
        ;
 TITLE1 " dasd reporting ";
 TITLE2 " &INQUNAME ";
 TITLE3 " &TMFRHDR ";
 TITLE4 " &FPTLBOTH ";
 LABEL SYSID="Sysid";
 LABEL ENDTIME="Time";
 FORMAT ENDTIME TIME8.;
 LABEL DEVADDR="Dev*Addr";
 LABEL VOLSER="Volser";
 LABEL DVAACTRT="Act*Rate";
 LABEL DVAAVRES="Resp*Time";
 FORMAT DVAAVRES 7.4;
 LABEL DVAAVIOS="IOSQ*Time";
 FORMAT DVAAVIOS 7.4;
 LABEL DVAAVPEN="Pending*Time";
 FORMAT DVAAVPEN 7.4;
 LABEL DVAUTL="Res.&Idle";
 FORMAT DVAUTL 7.4;
 LABEL DVAAVSER="Serv*Time";
 FORMAT DVAAVSER 7.4;
 LABEL DVAAVDIS="Discon*Time";
 FORMAT DVAAVDIS 7.4;
 LABEL DVASSC="Strt Subch*Count";
 FORMAT DVASSC 4.0;
 LABEL DVAAVCNN="Connect*Time";
 FORMAT DVAAVCNN 7.4;
 RUN;
 OPTIONS NOSOURCE NOSOURCE2 NOMACROGEN NOSYMBOLGEN NOMLOGIC
NONOTES;
 DATA _NULL_; CALL MICSLOG('ICF1099'); RUN;
àà
//FT11F001 DD SYSOUT=(*,,)
//FT12F001 DD DSN=K.X06601.D022A1.DATA,
//              DISP=(NEW,CATLG,CATLG),
```

If the SAS routines and the MICS process worked correctly, the

```
//              UNIT=SYSTSO,                required measures will be
//              SPACE=(TRK,(15,15)),        available in this dataset.
//              DCB=(LRECL=132,BLKSIZE=23364,RECFM=FB)
```

All the following description are resources necessary for the MICS report construction.

```
//LIBRARY DD DISP=(NEW,DELETE,DELETE),
//             UNIT=SYSDA,
//             DCB=(DSORG=PO,RECFM=U,BLKSIZE=9076),
//             SPACE=(TRK,(5,5,5),,,)
//MICSLOG DD SYSOUT=(*,,)
//PDETAIL DD DISP=(SHR,KEEP,KEEP),
//             DSN=Q2.SL.CA.MICS.PRSMF.DETAIL
//SORTCNTL DD DISP=(SHR,KEEP,KEEP),
//             DSN=S0.SL.CA.MICS.PARMS(DFSORT)
//SORTWK01 DD DISP=(NEW,DELETE,DELETE),
//             UNIT=SYSDA,
//             DCB=(BLKSIZE=9076),
//             SPACE=(CYL,(40,20,),,,)
//SORTWK02 DD DISP=(NEW,DELETE,DELETE),
//             UNIT=SYSDA,
//             DCB=(BLKSIZE=9076),
//             SPACE=(CYL,(40,20,),,,)
//SORTWK03 DD DISP=(NEW,DELETE,DELETE),
//             UNIT=SYSDA,
//             DCB=(BLKSIZE=9076),
//             SPACE=(CYL,(40,20,),,,)
//WORK DD DISP=(NEW,DELETE,DELETE),
//             UNIT=SYSDA,
//             DCB=(DSORG=DA,RECFM=U,BLKSIZE=9076),
//              SPACE=(CYL,(40,20,),,,)
//SASAUTOS DD DISP=(SHR,KEEP,KEEP),
//              DSN=S0.SL.CA.SAS.MACAUTOS
//          DD DISP=(SHR,KEEP,KEEP),
//              DSN=S0.SL.CA.MICS.PRSMF.USER.SOURCE
//          DD DISP=(SHR,KEEP,KEEP),
//              DSN=S0.SL.CA.MICS.MACAUTOS
//SASHELP DD DISP=(SHR,KEEP,KEEP),
//              DSN=S0.SL.CA.SAS.SASHELP
//STEPLIB DD DISP=(SHR,KEEP,KEEP),
//              DSN=S0.SL.CA.SAS.LIBRARY
//          DD DISP=(OLD,DELETE,DELETE),
//              DSN=*.LIBRARY,UNIT=SYSDA,VOL=REF=*.LIBRARY
//          DD DISP=(SHR,KEEP,KEEP),
//              DSN=S0.SL.CA.MICS.PRSMF.USER.LOAD
//          DD DISP=(SHR,KEEP,KEEP),
//              DSN=S0.SL.CA.MICS.LOAD
//SASMSGS DD DISP=(SHR,KEEP,KEEP),
```

```
//              DSN=S0.SL.CA.SAS.SASMSGS
//SASUTL DD DISP=(SHR,KEEP,KEEP),
//              DSN=S0.SL.CA.SAS.UTILLIB
//SOURCE DD DISP=(SHR,KEEP,KEEP),
//              DSN=S0.SL.CA.MICS.SOURCE
//USOURCE DD DISP=(SHR,KEEP,KEEP),
//              DSN=S0.SL.CA.MICS.PRSMF.USER.SOURCE
```

B) IMPLEMENTATION PROBLEM

A batch process, frequently called a job, is made of step(s). Each step required a certain amount of resources. All the resources needed for the job execution are reserved when the procedure is submitted. This means that if a specific resource, such as a dataset, must be updated at the end of a job. This resource will stay idle during the entire job execution, waiting to be updated.
We are quiet concerned about this kind of restriction because the mailbox dataset is updated by the batch process and read by the interactive one.
Both of them are totally independent and a contention occurs when a Batch process is executed, and either another Batch is submitted or the user enables the interactive part.

The mailbox dataset concept was thought with a resource reserved at the step level and not at the job level. As a solution, the dataset name cannot figure explicitly in the job procedure.
To pass round this restriction the last step consists to execute a routine unique by its name, updating the so called mailbox dataset implicitly. This way the mailbox dataset name does not figure in the job procedure explicitly, and no reserved can be made about it.

```
//ST002 EXEC PPTSOBAT,CLIB='K.X06601.Z50306,DISP=SHR',
//    PARM='%PPISPST CMD(PTRANS)'
//SYSTSIN DD DUMMY
//ST003 EXEC IDCAMS
 DEL K.X06601.Z50306
```

This is a sample of the mailbox dataset content. The first item identifies the day, the system, and the day-period of a set of required measures. The second item allows to identify the routine enabling the link between the batch and the interactive process.

```
022SYSAP1 Z50306
```

C) SAMPLE OF AN INTERACTIVE EXECUTION

The main application language used is REXX. REXX is a CLIST like language with many possibilities such as integrating some other languages in its own code. All the interactive manipulation have been developed with the ISPF interface.

The goal of this section consists to present the user interface via the tool panels.

The 'Test Tuning Dialogue' is available within the first ISPF main menu for space manager. The application is mainly splitted into five areas which are the thresholds specification menu, thresholds modify menu, the thresholds and results suppression menu, the display menu and the execution menu.

```
------------------------------- TSO/E  LOGON ---------------------------------
PF1/PF13 ==> Help    PF3/PF15 ==> Logoff    PA1 ==> Attention    PA2 ==> Reshow
You may request specific HELP information by entering a '?' in any entry field.
   ENTER LOGON PARAMETERS BELOW:                        RACF LOGON PARAMETERS:

   USERID    ===> X06601

   PASSWORD  ===>                                       NEW PASSWORD ===>

   PROCEDURE ===> PP02B                                 GROUP IDENT  ===>

   ACCT NMBR ===> 93406698

   SIZE      ===> 4096

   PERFORM   ===>

   COMMAND   ===>

   ENTER AN 'S' BEFORE EACH OPTION DESIRED BELOW:

           -NOMAIL         -NONOTICE         -RECONNECT        -OIDCARD

                                                                          O-OO1
```

```
ICH70001I X06601   LAST ACCESS AT 13:34:46 ON WEDNESDAY, JANUARY 23, 1991
IKJ56455I X06601 LOGON IN PROGRESS AT 13:39:47 ON JANUARY 23, 1991
IKJ56951I NO BROADCAST MESSAGES
***
```

O-001

```
PP02B --------------- ISPF MAIN MENU FOR SPACE MANAGERS -----------------------
OPTION  ===>
                                                               USERID   - X06601
   0  ISPF PARMS  - Specify terminal and user parameters       TIME     - 13:41
   1  PDF         - Program Development Facility               TERMINAL - 3278
   C  CHANGES     - Change management                          PF KEYS  - 24
   CO CONTACT     - Soft-Switch Electronic mail
   D  PRINT       - Print on remote printer
   E  EREP        - EREP Dialog
   F  FLASHER     - Browse output on SPOOL
   I  ISMF        - Interactive Storage Management Facility
   L  LIBRARIAN   - Librarian processor
   M  MAIL        - Message communication  (electronic mail)
   N  NEWS        - Daily news
   P  PROD        - Production products
   R  REPORT      - Production incident report entry
   S  MICS        - Management Information Control System
   T  TELEF       - Telephone numbers
   V  VPW         - Examine VPW data sets (Please use VO for old version)
   Z  MVS/DITTO   - Utility program
   TU Tuning      - Test TUNING dialog.
   X  EXIT        - Terminate ISPF using log and list defaults

Enter END command to terminate application.
```

O-001

```
-------------------- TUNING Dialog Main Menu ----------------------------------
                     -----------------------
 COMMAND ===>
                                                              USERID: X06601
                                                              TIME: 13:42


            1 Thresholds specification menu
            2 Thresholds modify menu
            3 Thresholds and results suppression menu
            4 Display menu

            5 Executions menu


            X End
-------------------------------------------------------------------------------
```

O-001

The thresholds specification menu enables the user to create either a general or an interval threshold. A general threshold may be created if it has not already been created for the specified device(s). An interval threshold may be created if a general threshold already exists and if it doesn't overpass an already existing interval threshold for the specified device(s).

```
-------------------- Thresholds Specification Menu ----------------------------
                     -----------------------------
 COMMAND ===>
                                                            USERID: X06601
                                                            TIME: 13:42



           1 General Thresholds Creation Menu
           2 Interval Thresholds Creation Menu




           X Return
--------------------------------------------------------------------------------

                                                                          O-001
```

```
-------------------- General Thresholds Creation -----------------------------
                     ----------------------------
 COMMAND ===>
                                                            USERID: X06601
                                                            TIME: 13:43



            Volser     :  atr*..  (*, BBB*, BBB215)
            System     :  a       (*, A, B)
            Period     :  1       (* : 1 & 2
                                   1 : 8:00 - 17:00
                                   2 : 17:00 - 8:00)



--------------------------------------------------------------------------------

                                                                          O-001
```

```
-------------------- General Thresholds Creation -----------------------------
                     ---------------------------
COMMAND ===>
                                                             USERID: X06601
                                                             TIME: 13:44



          Activity Rate           :  0..
          I/O Response Time       :  30.
          Queue Time              :  0..
          Pending Time            :  0..
          Reserved & Idle         :  0..
          Service Time            :  0..
          Disconnect Time         :  0..
          Strt Subchannel Count   :  0..
          Connect Time            :  0..

-------------------------------------------------------------------------------

                                                                        0-001
```

```
-------------------- Interval Thresholds Creation ----------------------------
                     ----------------------------
COMMAND ===>
                                                             USERID: X06601
                                                             TIME: 13:46



            Volser       :  ATR*..   (*, BBB*, BBB215)
            System       :  A        (*, A, B)
            Period       :  1        (1 : 8:00 - 17:00
                                      2 : 17:00 - 8:00)



-------------------------------------------------------------------------------

                                                                        0-001
```

```
-------------------- Interval Thresholds Creation ----------------------------
                     ----------------------------
COMMAND ===>
                                                             USERID: X06601
                                                             TIME: 13:51


     Start point    :  1300
     End point      :  1400
          Activity Rate           :  0..
          I/O Response Time       :  25.
          Queue Time              :  0..
          Pending Time            :  0..
          Reserved & Idle         :  0..
          Service Time            :  0..
          Disconnect Time         :  0..
          Strt Subchannel Count   :  0..
          Connect Time            :  0..

-------------------------------------------------------------------------------

                                                                        0-001
```

The thresholds modify menu enables the user to change the components of existing threshold(s) for the specified device(s).

```
-------------------- Thresholds Modify Menu ----------------------------------
                     ----------------------
COMMAND ===>
                                                             USERID: X06601
                                                             TIME: 13:52



           1 General Thresholds Modification
           2 Interval Thresholds Modification




           X Return
------------------------------------------------------------------------------

                                                                      O-001
```

```
-------------------- General Thresholds Modification -------------------------
                     -------------------------------
COMMAND ===>
                                                             USERID: X06601
                                                             TIME: 13:53



            Volser      :  ATR*..   (*, BBB*, BBB215)
            System      :  A        (*, A, B)
            Period      :  1        (* : 1 & 2
                                     1 : 8:00 - 17:00
                                     2 : 17:00 - 8:00)



------------------------------------------------------------------------------

                                                                      O-001
```

```
-------------------- General Thresholds Modification -------------------------
                     -------------------------------
COMMAND ===>
                                                         USERID: X06601
                                                         TIME: 13:54



          Activity Rate          :  0..
          I/O Respons Time       :  30.
          Queue Time             :  0..
          Pending Time           :  0..
          Reserved & Idle        :  0..
          Service Time           :  0..
          Disconnect Time        :  0..
          Strt Subchannel Count  :  0..
          Connect Time           :  0..

------------------------------------------------------------------------------

                                                                        O-001
```

```
-------------------- Interval Thresholds Modification ------------------------
                     --------------------------------
COMMAND ===>
                                                         USERID: X06601
                                                         TIME: 13:54



          Volser     :  ATR*..  (*, BBB*, BBB215)
          System     :  A       (*, A, B)
          Period     :  1       (* : 1 & 2
                                 1 : 8:00 - 17:00
                                 2 : 17:00 - 8:00)



------------------------------------------------------------------------------

                                                                        O-001
```

```
-------------------- Interval Thresholds Modification ------------------------
                     --------------------------------
COMMAND ===>
                                                         USERID: X06601
                                                         TIME: 13:55


     Start point    :  900
     End point      :  1200
          Activity Rate          :  0..
          I/O Response Time      :  32.
          Queue Time             :  0..
          Pending Time           :  0..
          Reserved & Idle        :  0..
          Service Time           :  0..
          Disconnect Time        :  0..
          Strt Subchannel Count  :  0..
          Connect Time           :  0..

------------------------------------------------------------------------------

                                                                        O-001
```

The execution menu is made of two parts. The first one enables the user to create a batch procedure to collect information via MICS, and to submit this procedure according to the adequate introduced parameters. The second one enables the user to compare both the thresholds set and the MICS collected data.

```
-------------------- Executions Menu -----------------------------------------
                     ---------------
COMMAND ===>
                                                          USERID: X06601
                                                          TIME: 13:57



            1 Batch Execution of Mics
            2 Thresholds Execution on Mics Results




            X Return
------------------------------------------------------------------------------

                                                                       0-001
```

```
-------------------- Batch Execution of Mics ---------------------------------
                     -----------------------
COMMAND ===>
                                                          USERID: X06601
                                                          TIME: 13:57



             Date      :  22       ( 1 - 366)
             System    :  A        (*, A, B)
             Period    :  1        (* : 1 & 2
                                    1 : 8:00 - 17:00
                                    2 : 17:00 - 8:00)



------------------------------------------------------------------------------

                                                                       0-001
```

```
----------------------------- TSO SUBMIT PANEL -----------------------------
     This panel is designed to create JOB and OUTPUT cards for you .
     If you do not want to use this facility , please override here
        with an  N  character :
     JOB and OUTPUT cards automatically created ====>  Y

     ENTER or VERIFY the following panel-variables ( only if Y above ) .

   //   JOBCARD
        Jobname Character(s)    ====>  23
        Account Number          ====>  93406600
        Execution Class         ====>  1               See help (PFK1)
        Userid                  ====>  X06601          Default X06601
        Password                ====>

   /*   XEQ CARD
        Execution node          ====>  BRU             NOS , BRU

   //   OUTPUT CARD
        Sysout Destination      ====>  BRU             NOS , BRU , LOCAL
        Writer                  ====>  U240            Uxxx
        Groupid                 ====>  D066SPAC        Dsername

  Press ENTER to submit , PF03 or PF15 to terminate , PF01 or PF13 for help
                                                                      O-OO1
```

```
------------------- Thresholds Execution on Mics Results ---------  ROW 1 OF 5
CMD : ===>                                                  SCROLL ===> CUR
                                                             DATE: 91/01/23
                                                                   91.023

CMD DATE SYS PER                          S  - select occurrence
-------------------------------------------------------------------------------
 .  014 SYSB P1
 .  015 SYSB P1
 .  016 SYSB P1
 .  017 SYSB P1
 .  022 SYSB P1
****************************** BOTTOM OF DATA ********************************





                                                                      O-OO1
```

The thresholds and results suppression menu is made of four parts. The two first one enables the user to delete either a general or an interval threshold. The two last one concerned the suppression of either the collected MICS measures or the compared report.

```
-------------------- Thresholds and Results Suppression ----------------------
                     ----------------------------------
COMMAND ===>
                                                              USERID: X06601
                                                              TIME: 14:01



           1 General Thresholds Suppression
           2 Interval Thresholds Suppression
           3 Mics Batch Output Suppression
           4 Thresholds Results Suppression



           X Return
-------------------------------------------------------------------------------

                                                                      O-OO1
```

```
-------------------- General Thresholds Suppression --------------------------
                     ------------------------------
COMMAND ===>
                                                              USERID: X06601
                                                              TIME: 14:02



            Volser     :  ATR*..  (*, BBB*, BBB215)
            System     :  A       (*, A, B)
            Period     :  1       (* : 1 & 2
                                   1 : 8:00 - 17:00
                                   2 : 17:00 - 8:00)



-------------------------------------------------------------------------------

                                                                      O-OO1
```

```
-------------------- Interval Thresholds Suppression -------------------------
                     -------------------------------
 COMMAND ===>
                                                          USERID: X06601
                                                          TIME: 14:03



             Volser      :  ATR*..  (*, BBB*, BBB215)
             System      :  A       (*, A, B)
             Period      :  1       (* : 1 & 2
                                     1 : 8:00 - 17:00
                                     2 : 17:00 - 8:00)




-------------------------------------------------------------------------------

                                                                         O-001
```

```
------------------ Thresholds REsults Suppression ----------------  ROW 1 OF 8
CMD : ===>                                                SCROLL ===> CUR
                                                           DATE: 91/01/23
                                                                 91.023

CMD DATE SYS PER                        S  - select occurrence
-------------------------------------------------------------------------------
 .  014 SYSA P1
 .  015 SYSA P1
 .  016 SYSA P1
 .  017 SYSA P1
 .  018 SYSA P1
 .  018 SYSB P1
 .  021 SYSA P1
 .  022 SYSA P1
****************************** BOTTOM OF DATA ********************************

                                                                         O-001
```

```
----------------- Mics Batch Output Suppression -----------------  ROW 1 OF 6
CMD : ===>                                                SCROLL ===> CUR
                                                           DATE: 91/01/23
                                                                 91.023

CMD DATE SYS PER                        S  - select occurrence
-------------------------------------------------------------------------------
 .  014 SYSB P1
 .  015 SYSB P1
 .  016 SYSB P1
 .  017 SYSB P1
 s  022 SYSA P1
 .  022 SYSB P1
****************************** BOTTOM OF DATA ********************************

                                                                         O-001
```

Each compared report is identified by a unique occurrence of Date, System, Period item. The display menu is made of five parts. The two first one enables the user to display the compared report(s). As the user required, the displayed report(s) may be sorted on any of its components.

The third one displays the thresholds set.

The two last one enables the user to see which compared report(s) are available.

```
-------------------- Display Menu -------------------------------------------
                     ------------
COMMAND ===>
                                                           USERID: X06601
                                                           TIME: 14:08



           Thresholds Result Display
              1 For one given occurence of (DATE SYS PER)
              2 For many occurences of (DATE SYS PER)

           3 Thresholds Display
           4 Mics Batch Parameters List
           5 Thresholds Results Parameters List

           X Return
-----------------------------------------------------------------------------

                                                                      O-OO1
```

```
-------------------- Sort Key Panel -----------------------------------------
                     --------------
COMMAND ===>
                                                           USERID: X06601
                                                           TIME: 14:16

          Existing Threshold      :  ..          Specify the sort order
          Volser                  :  ..          for the following fields
          Time                    :  ..          (' ',1..21)
                                            Threshold Type
          Activity Rate           :  ..          : ..
          I/O Response Time       :  ..          : ..
          Queue Time              :  ..          : ..
          Pending Time            :  ..          : ..
          Reserved & Idle         :  ..          : ..
          Service Time            :  ..          : ..
          Disconnect Time         :  ..          : ..
          Strt Subchannel Count   :  ..          : ..
          Connect Time            :  ..          : ..

-----------------------------------------------------------------------------

                                                                      O-OO1
```

```
-------------------- Sort Key Panel ------------------------------------------
                     --------------
COMMAND ===>
                                                        USERID: X06601
                                                        TIME: 14:16

          Existing Threshold     :  3.        Specify the sort order
          Volser                 :  1.        for the following fields
          Time                   :  2.        (' ',1..21)
                                          Threshold Type
          Activity Rate          :  ..        : ..
          I/O Response Time      :  ..        : ..
          Queue Time             :  ..        : ..
          Pending Time           :  ..        : ..
          Reserved & Idle        :  ..        : ..
          Service Time           :  ..        : ..
          Disconnect Time        :  ..        : ..
          Strt Subchannel Count  :  ..        : ..
          Connect Time           :  ..        : ..

-------------------------------------------------------------------------------

                                                                         0-001
```

```
----------------- Thresholds Result Display -------------------- ROW 1 OF 7
                                             For one given occurrence
CMD : ===>                                                SCROLL ===> CUR
                                                           DATE: 91/01/23
                                                                 91.023

CMD DATE SYS PER                          S  - select occurrence
-------------------------------------------------------------------------------
 .  014 SYSA P1
 .  015 SYSA P1
 .  016 SYSA P1
 .  017 SYSA P1
 .  018 SYSA P1
 .  018 SYSB P1
 s  021 SYSA P1
****************************** BOTTOM OF DATA ********************************

                                                                         0-001
```

Mics Batch Parameters List

```
-------------------- SORT KEY PANEL ------------------------------------------
                     --------------
 COMMAND ===>
                                                                USERID: X06601
           Date               :  .                              TIME: 14:43

           System             :  .               Specify the sort order
           Period             :  .               for the following fields
                                                 (' ',1..3)

------------------------------------------------------------------------------
                                                                         0-001
```

```
EDIT ---- K.X06601.TAMPON.DATA ------------------------------ COLUMNS 001 0'
COMMAND ===>                                                  SCROLL ===> CSR
      DATE SYS PER
****** ***************************** TOP OF DATA ******************************
000001 014 SYSB P1 1
000002 015 SYSB P1 1
000003 016 SYSB P1 1
000004 017 SYSB P1 1
000005 022 SYSB P1 1
****** **************************** BOTTOM OF DATA ****************************
                                                                         0-001
```

```
EDIT ---- K.X06601.TAMPON.DATA ------------------------------ COLUMNS 001 072
COMMAND ===>                                                  SCROLL ===> CSR
      DATE SYS PER
****** ***************************** TOP OF DATA ******************************
000001 014 SYSA P1 2
000002 015 SYSA P1 2
000003 016 SYSA P1 2
000004 017 SYSA P1 2
000005 018 SYSA P1 2
000006 018 SYSB P1 2
000007 021 SYSA P1 2
000008 022 SYSA P1 2
****** **************************** BOTTOM OF DATA ****************************
                                                                         0-001
```

Annexe 4.A

By this Annexe, the Interactive Storage Management Facility (ISMF) will be roughly illustrated. The goal of thissection is to provide to the reader a practical idea of the SMS interface.

```
PP2ON --------------- ISPF MAIN MENU FOR PROD in   BUC    ---------------------
OPTION  ===> _
                                                         USERID   - X06601
   0  ISPF PARMS  - Specify terminal and user parameters  TIME     - 10:27
   1  PDF         - Program Development Facility          TERMINAL - 3278
   B  IMF         - Boole & Babbage IMS Managt Facility   PF KEYS  - 24
   D  PRINT       - Print on remote printer
   E  EREP        - EREP Dialog
   F  FLASHER     - Browse output on SPOOL
   I  ISMF        - Interactive Storage Management Facility
   L  LIBRARIAN   - Librarian processor
   N  NEWS        - Daily news
   P  PROD        - Production products
   Z  MVS/DITTO   - Utility program
   X  EXIT        - Terminate ISPF using log and list defaults

Enter END command to terminate application.
```

```
                    ISMF PRIMARY OPTION MENU
ENTER SELECTION OR COMMAND ===> _

SELECT ONE OF THE FOLLOWING:

 0  ISMF PROFILE        - Change ISMF user profile
 1  DATA SET            - Perform Functions Against Data Sets
 2  VOLUME              - Perform Functions Against Volumes
 3  MANAGEMENT CLASS    - Display Backup and Migration Criteria
 4  DATA CLASS          - Display Data Set Allocation Parameters
 5  STORAGE CLASS       - Display Performance and Availability Criteria
 X  EXIT                - Terminate ISMF
```

```
                    DATA SET SELECTION ENTRY PANEL              Page 1 of 4
COMMAND ===> _

FOR A DATA SET LIST, SELECT SOURCE OF GENERATED LIST  ===> 2   (1 OR 2)

  1  GENERATE FROM A SAVED LIST
       LIST NAME  ===>

  2  GENERATE A NEW LIST FROM CRITERIA BELOW
       DATA SET NAME    ===> **

       SPECIFY SOURCE OF THE NEW LIST  ===> 2    (1 - VTOC, 2 - Catalog)

       1 GENERATE LIST FROM VTOC
          VOLUME SERIAL NUMBER  ===>             (fully or partially specified)
       2 GENERATE LIST FROM CATALOG
          CATALOG NAME  ===>
            CATALOG PASSWORD    ===>             (if password protected)
          VOLUME SERIAL NUMBER  ===>             (fully or partially specified)
          ACQUIRE DATA FROM VOLUME         ===> Y  (Y or N)
          ACQUIRE DATA IF DFHSM MIGRATED   ===> Y  (Y or N)

USE ENTER TO PERFORM SELECTION; USE DOWN COMMAND TO VIEW NEXT SELECTION PANEL;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

These panels enable the users to know features about datasets

```
                         DATA SET LIST
COMMAND ===> _                                    SCROLL ===> HALF
                                              Entries 1-9 of 9
ENTER LINE OPERATORS BELOW:                   Data Columns 3-7 of 31
```

LINE OPERATOR	DATA SET NAME	ALLOC SPACE	ALLOC USED	% NOT USED	NUM EXT	ALLOC UNIT	SEC ALLOC	DS ORG	REC FMT	RECORD LENGTH	BLK SZ CI SIZE	OPTIMAL SIZE	BLOCK UNUSED	VOLUME SERIAL
---(1)----	------------(2)------------	--(3)--	--(4)--	-(5)-	(6)	-(7)-	--(8)--	(9)	(10)-	-(11)-	-(12)--	-(13)--	-(14)--	-(15)-
	L.U59497.$$MAIL$$.DATA	4872	4684	3	2	BLK	234	PS	FB	80	4000	23440	672	DEVC00
	L.U59497.IEBDG.T1	47	-------	---	---	---	-------	PS	FB	-----	23440	-----	-------	MIGRAT
	L.U59497.IEBDG.T3	47	-------	---	---	---	-------	PS	FB	-----	23440	-----	-------	MIGRAT
	L.U59497.IPCS.DIRECTRY.CD	703	-------	---	1	CYL	703	VS	U	4089	4096	-----	-------	DEVC01
	L.U59497.IPCS.DIRECTRY.CI	47	-------	---	1	TRK	47	VS	U	2041	2048	-----	-------	DEVC01
	L.U59497.IPCS.DIRECTRY.CX	-------	-------	---	---	---	-------	VS	-----	-----	-----	-----	-------	------
	L.U59497.ISMF.CNTL	47	-------	---	---	---	-------	PS	FB	-----	6160	-----	-------	MIGRAT
	L.U59497.LIB.CNTL	47	-------	---	---	---	-------	PO	FB	-----	6160	-----	-------	MIGRAT
	L.U59497.SPFLOG2.LIST	47	-------	---	---	---	-------	PS	VA	-----	129	-----	-------	MIGRAT

---------- ------ ----------- BOTTOM OF DATA ----------- ------ ---- ----------- ------ ---- ----------- ------ ----

DATA SET LIST

COMMAND ===> _ SCROLL ===> HALF

Entries 1-9 of 9

ENTER LINE OPERATORS BELOW: Data Columns 17-19 of 31

LINE OPERATOR	DATA SET NAME	DEVICE TYPE	CREATE DATE	EXPIRE DATE	LAST REF DATE	LAST BACKUP DATE	CHG IND	DATA CLASS NAME	MANAGEMENT CLASS NAME	STORAGE CLASS NAME	DATA SET ENVIRONMENT
---(1)----	------------(2)------------	-(17)--	---(18)---	---(19)---	---(20)---	---(21)----	(22)	---(23)---	---(24)---	---(25)---	---(27)----
	L.U59497.$$MAIL$$.DATA	3380	1989/05/18	0000/00/00	1991/05/13	1991/05/09	YES	--------	MCDEV	SCSTD	MANAGED
	L.U59497.IEBDG.T1	3380	1991/04/26	0000/00/00	1991/04/26	----------	NO	--------	MCDEV	SCSTD	MANAGED
	L.U59497.IEBDG.T3	3380	1991/04/26	0000/00/00	1991/04/26	----------	NO	SEQ80	MCDEV	SCSTD	MANAGED
	L.U59497.IPCS.DIRECTRY.CD	3380	1990/10/05	0000/00/00	1991/05/13	----------	YES	--------	--------	--------	---------
	L.U59497.IPCS.DIRECTRY.CI	3380	1990/10/05	0000/00/00	0000/00/00	----------	NO	--------	--------	--------	---------
	L.U59497.IPCS.DIRECTRY.CX	-------	1990/10/05	0000/00/00	----------	1991/05/08	---	--------	MCDEV	SCSTD	MANAGED
	L.U59497.ISMF.CNTL	3380	1989/08/22	0000/00/00	1991/01/02	----------	NO	--------	MCDEV	SCSTD	MANAGED
	L.U59497.LIB.CNTL	3380	1989/10/20	0000/00/00	1991/02/25	----------	NO	--------	MCDEV	SCSTD	MANAGED
	L.U59497.SPFLOG2.LIST	3380	1991/05/06	0000/00/00	1991/05/07	----------	YES	--------	MCBATCHA	SCSTD	MANAGED

---------- ------ ----------- BOTTOM OF DATA ----------- ------ ---- ----------- ------ ---- ----------- ------ ---- ------ ----

```
                    VOLUME SELECTION ENTRY PANEL              Page 1 of 3
COMMAND ===> _

SELECT SOURCE TO GENERATE VOLUME LIST  ===> 2   (1 - Saved list, 2 - New list)

   1  GENERATE FROM A SAVED LIST
        LIST NAME  ===>
   2  GENERATE A NEW LIST FROM CRITERIA BELOW
        SPECIFY SOURCE OF THE NEW LIST  ===> 2  (1 - Physical, 2 - SMS)

        OPTIONALLY SPECIFY ONE OR MORE:
          TYPE OF VOLUME LIST    ===> 1       (1-Online,2-Not Online,3-Either)
          VOLUME SERIAL NUMBER   ===> TST*    (fully or partially specified)
          DEVICE TYPE            ===>         (fully or partially specified)
          DEVICE NUMBER          ===>         (fully specified)
            TO DEVICE NUMBER     ===>         (for range of devices)
          ACQUIRE PHYSICAL DATA  ===> Y       (Y or N)
          ACQUIRE SPACE DATA     ===> Y       (Y or N)
          STORAGE GROUP NAME     ===> *       (fully or partially specified)
          CDS NAME  ===> 'ACTIVE'
                                              (fully specified or 'ACTIVE')
```

These panels enable the storage administrator to know features about volumes

VOLUME LIST

COMMAND ===> _ SCROLL ===> HALF

Entries 1-10 of 10

ENTER LINE OPERATORS BELOW: Data Columns 3-8 of 39

LINE OPERATOR	VOLUME SERIAL	FREE SPACE	% FREE	ALLOC SPACE	FRAG INDEX	LARGEST EXTENT	FREE EXTENTS	INDEX STATUS	FREE DSCBS	FREE VIRS	DEVICE TYPE	DEV NUM	SHR DASD	USE ATTR
---(1)----	-(2)--	--(3)--	(4)-	--(5)--	-(6)-	--(7)--	--(8)--	--(9)---	(10)-	(11)-	-(12)--	(13)	(14)	(15)
	TSTA00	206440	33	415411	389	58320	90	ENABLED	1155	223	3380	C47	YES	PRIV
	TSTA01	182972	29	438879	561	25998	227	ENABLED	1029	218	3380	C49	YES	PRIV
	TSTA02	172619	28	449232	533	56353	295	ENABLED	1595	207	3380	CA0	YES	PRIV
	TSTA03	170324	27	451527	587	18972	283	ENABLED	1723	213	3380	CA1	YES	PRIV
	TSTA04	150228	24	471623	606	12320	226	ENABLED	1914	219	3380	C43	YES	PRIV
	TSTA05	173228	28	448623	570	13538	209	ENABLED	1863	209	3380	C46	YES	PRIV
	TSTA06	161705	26	460146	536	28528	189	ENABLED	1986	221	3380	C45	YES	PRIV
	TSTA07	180208	29	441643	619	11805	261	ENABLED	1905	215	3380	CA3	YES	PRIV
	TSTC02	581003	31	1284549	342	122356	37	ENABLED	3894	515	3380	87F	YES	PRIV
	TSTC03	599975	32	1265577	326	128539	34	ENABLED	3911	515	3380	869	YES	PRIV

---------- ------ ----------- BOTTOM OF DATA ----------- ------ ----------- BOTTOM OF DATA ----------- ------

VOLUME LIST

COMMAND ===> _ SCROLL ===> HALF

Entries 1-10 of 10

ENTER LINE OPERATORS BELOW: Data Columns 19-24 of 39

LINE OPERATOR	VOLUME SERIAL	DUPLEX STATUS	OTHER DEVICE	SUBSYS ID	PHYSICAL STATUS	STORAGE GRP NAME	SYSC SMS
---(1)----	-(2)--	--(19)--	-(20)-	-(21)-	--(22)--	--(23)--	--(24)--
	TSTA00	NONE	---	----	CONVERT	SGTEST	ENABLE
	TSTA01	NONE	---	----	CONVERT	SGTEST	ENABLE
	TSTA02	NONE	---	----	CONVERT	SGTEST	ENABLE
	TSTA03	NONE	---	----	CONVERT	SGTEST	ENABLE
	TSTA04	NONE	---	----	CONVERT	SGTEST	ENABLE
	TSTA05	NONE	---	----	CONVERT	SGTEST	ENABLE
	TSTA06	NONE	---	----	CONVERT	SGTEST	ENABLE
	TSTA07	NONE	---	----	CONVERT	SGTEST	ENABLE
	TSTC02	SIMPLEX	---	00C2	CONVERT	SGTESTL	ENABLE
	TSTC03	SIMPLEX	---	00C2	CONVERT	SGTESTL	ENABLE

---------- ------ ----------- BOTTOM OF DATA ----------- ------ ----

```
                 MANAGEMENT CLASS APPLICATION SELECTION
COMMAND ===> _

TO PERFORM MANAGEMENT CLASS OPERATIONS, SPECIFY:

  CDS NAME               ===> 'ACTIVE'
                                  (1 to 44 character data set name or 'ACTIVE')

  MANAGEMENT CLASS NAME ===> *              (For Management Class List, fully or
                                             partially specified or * for all)

SELECT ONE OF THE FOLLOWING OPTIONS ===> 1

  1  LIST    - Generate a list of Management Classes
  2  DISPLAY - Display a Management Class

IF OPTION 1 CHOSEN ABOVE,
   RESPECIFY SORT CRITERIA          ===> N    (Y or N)
```

These panels enable the storage administrator to know features about Management Class

MANAGEMENT CLASS LIST

COMMAND ===> _ SCROLL ===> HALF

Entries 1-11 of 16

Data Columns 3-7 of 32

CDS NAME: ACTIVE

ENTER LINE OPERATORS BELOW:

LINE OPERATOR	MGMTCLAS NAME	EXPIRE NON-USAGE	EXPIRE DATE/DAYS	RET LIMIT	PARTIAL RELEASE	PRIMARY DAYS	LEVEL 1 DAYS	CMD/AUTO MIGRATE	# GDG ON PRIMARY	ROLLED-OFF GDS ACTION
---(1)----	--(2)---	---(3)---	---(4)----	--(5)--	----(6)----	---(7)--	--(8)--	--(9)---	--(10)--	---(11)---
	MCBATCH	40	NOLIMIT	0	YES	2	0	BOTH	---	EXPIRE
	MCBATCHA	10	NOLIMIT	0	YES	2	0	BOTH	---	EXPIRE
	MCBATCHB	15	NOLIMIT	0	YES	2	0	BOTH	---	EXPIRE
	MCBATCHC	20	NOLIMIT	0	YES	2	0	BOTH	---	EXPIRE
	MCDEV	190	NOLIMIT	0	YES	4	20	BOTH	---	EXPIRE
	MCIC1	190	NOLIMIT	0	YES	5	25	BOTH	---	EXPIRE
	MCIC11	190	NOLIMIT	0	YES	5	0	BOTH	---	EXPIRE
	MCIC2	100	NOLIMIT	0	YES	10	30	BOTH	---	EXPIRE
	MCIC21	100	NOLIMIT	0	YES	10	0	BOTH	---	EXPIRE
	MCIC3	NOLIMIT	NOLIMIT	0	NO	10	35	BOTH	---	-------
	MCLIB	NOLIMIT	NOLIMIT	0	NO	30	60	BOTH	---	EXPIRE

MANAGEMENT CLASS LIST

COMMAND ===> _

CDS NAME: ACTIVE

ENTER LINE OPERATORS BELOW:

LINE OPERATOR	MGMTCLAS NAME	BACKUP FREQUENCY	# BACKUPS (DS EXISTS)	# BACKUPS (DS DELETED)	RETAIN DAYS ONLY BACKUP	RETAIN DAYS EXTRA BACKUPS	ADM/USER BACKUP	AUTO BACKUP	LAST MOD USERID	LAST DATE MODIFIED
---(1)----	--(2)---	--(12)---	---(13)----	----(14)----	---(15)----	----(16)-----	--(17)--	-(18)-	--(19)--	---(20)---
	MCBATCH	9999	--	--	-------	-------	NONE	NO	U48201	1991/04/30
	MCBATCHA	9999	--	--	-------	-------	NONE	NO	U48201	1991/04/30
	MCBATCHB	9999	--	--	-------	-------	NONE	NO	U48201	1991/04/30
	MCBATCHC	9999	--	--	-------	-------	NONE	NO	U48201	1991/04/30
	MCDEV	0	3	1	90	NOLIMIT	BOTH	YES	U48201	1990/10/04
	MCIC1	0	3	1	250	NOLIMIT	BOTH	YES	U64019	1990/09/25
	MCIC11	0	3	1	250	NOLIMIT	BOTH	YES	U48201	1991/02/06
	MCIC2	0	1	1	400	NOLIMIT	BOTH	YES	U64019	1990/09/25
	MCIC21	0	1	1	400	NOLIMIT	BOTH	YES	U48201	1991/02/06
	MCIC3	0	1	1	60	NOLIMIT	BOTH	YES	U48201	1990/10/10
	MCLIB	0	3	1	400	NOLIMIT	BOTH	YES	U48201	1991/02/14

```
                      DATA CLASS APPLICATION SELECTION
COMMAND ===> _

TO PERFORM DATA CLASS OPERATIONS, SPECIFY:

  CDS NAME        ===> 'ACTIVE'
                                 (1 to 44 character data set name or 'ACTIVE')

  DATA CLASS NAME ===> *                   (For Data Class List, fully or
                                            partially specified or * for all)

SELECT ONE OF THE FOLLOWING OPTIONS  ===> 1

 1   LIST    - Generate a list of Data Classes
 2   DISPLAY - Display a Data Class

IF OPTION 1 CHOSEN ABOVE,
   RESPECIFY SORT CRITERIA           ===> N   (Y or N)
```

These panels enable the storage administrator to know features about Data Class

DATA CLASS LIST

COMMAND ===> _ SCROLL ===> HALF

Entries 1-3 of 3

Data Columns 3-9 of 25

CDS NAME: ACTIVE

ENTER LINE OPERATORS BELOW:

LINE OPERATOR	DATACLAS NAME	RECORG	RECFM	LRECL	KEYLEN	KEYOFF	AVGREC	AVG VALUE	SPACE PRIMARY	SPACE SECONDARY	SPACE DIRECTORY	RETPD OR EXPDT
---(1)----	--(2)---	-(3)--	-(4)-	-(5)-	-(6)--	-(7)--	-(8)--	-(9)-	-(10)--	--(11)---	--(12)---	---(13)---
	PDS80	--	FB	80	---	-----	U	80	500	2500	12	----------
	SEQV255	--	VB	255	---	-----	U	255	180	1800	------	----------
	SEQ80	--	FB	80	---	-----	U	80	500	2500	------	----------

---------- -------- ------ BOTTOM OF DATA ------ -------- ---------- ------ -------- ----------

```
                  STORAGE CLASS APPLICATION SELECTION
COMMAND ===> _

TO PERFORM STORAGE CLASS OPERATIONS, SPECIFY:

  CDS NAME           ===> 'ACTIVE'
                                       (1 to 44 character data set name or 'ACTIVE')

  STORAGE CLASS NAME ===> *                    (For Storage Class List, fully or
                                                partially specified or * for all)

SELECT ONE OF THE FOLLOWING OPTIONS  ===> 1

 1   LIST    - Generate a list of Storage Classes
 2   DISPLAY - Display a Storage Class


IF OPTION 1 CHOSEN ABOVE,
   RESPECIFY SORT CRITERIA           ===> N  (Y or N)
```

These panels enable the storage administrator to know features about Storage Class

STORAGE CLASS LIST

COMMAND ===> _ SCROLL ===> HALF

Entries 1-8 of 8

Data Columns 3-8 of 13

CDS NAME: ACTIVE

ENTER LINE OPERATORS BELOW:

LINE OPERATOR	STORCLAS NAME	DIR RESP (MSEC)	DIR BIAS	SEQ RESP (MSEC)	SEQ BIAS	AVAIL- ABILITY	GUARANTEED SPACE	LAST MOD USERID	LAST DATE MODIFIED	LAST TIME MODIFIED
---(1)----	--(2)---	--(3)---	(4)-	--(5)---	(6)-	---(7)----	---(8)----	--(9)---	---(10)---	--(11)---
	SCGMUCA	5	-	5	-	STANDARD	YES	U48201	1990/10/09	11:30
	SCGUARAN	20	-	20	-	STANDARD	YES	U48201	1990/07/04	14:06
	SCMUCACH	5	-	5	-	STANDARD	NO	U48201	1990/10/31	12:13
	SCNONE	---	-	---	-	STANDARD	NO	U48201	1990/03/28	14:44
	SCNVCACH	999	-	999	-	STANDARD	NO	U48201	1990/07/04	14:26
	SCNVIO	20	-	20	-	STANDARD	NO	U48201	1990/07/04	14:06
	SCSTD	20	-	20	-	STANDARD	NO	U48201	1990/07/04	14:06
	SC23CACH	15	-	15	-	STANDARD	NO	U48201	1990/07/04	14:10

---------- -------- ------ BOTTOM OF DATA ------ -------- ---------- ------ --------